150 ptas.



CURSO PRACTICO DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR

Editorial Delta, S.A.

MI COMPUTER CURSO PRACTICO DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR

Publicado por Editorial Delta, S.A., Barcelona, y comercializado en exclusiva por Distribuidora Olimpia, S.A., Barcelona

Volumen III - Fascículo 28

Director:	José Mas Godayol
Director editorial:	Gerardo Romero
Jefe de redacción:	Pablo Parra
Coordinación editorial:	Jaime Mardones
Asesor técnico:	Francisco Martín Jesús Nebra

Redactores y colaboradores: G. Jefferson, R. Ford, S. Tarditti. A. Cuevas, F. Blasco

Para la edición inglesa: R. Pawson (editor), D. Tebbutt (consultant editor), C. Cooper (executive editor), D. Whelan (art editor), Bunch Partworks Ltd. (proyecto y realización)

Realización gráfica: Luis F. Balaguer

Redacción y administración:
Paseo de Gracia, 88, 5.°, Barcelona-8
Tels. (93) 215 10 32 / (93) 215 10 50 - Télex 97848 EDLTE

MI COMPUTER, *Curso práctico del ordenador personal, el micro y el miniordenador,* se publica en forma de 96 fascículos de aparición semanal, encuadernables en ocho volúmenes. Cada fascículo consta de 20 páginas interiores y sus correspondientes cubiertas. Con el fascículo que completa cada uno de los volúmenes, se ponen a la venta las tapas para su encuadernación.
El editor se reserva el derecho de modificar el precio de venta del fascículo en el transcurso de la obra, si las circunstancias del mercado así lo exigieran.

ISBN: 84-85822-83-8 (fascículo) 84-85822-94-3 (tomo 3)
84-85822-82-X (obra completa)
Depósito Legal: B. 52/1984

Fotocomposición: Tecfa, S.A., Pedro IV, 160, Barcelona-5
Impresión: Cayfosa, Santa Perpètua de Mogoda (Barcelona) 258407
Impreso en España - Printed in Spain - Julio 1984

Editorial Delta, S.A., garantiza la publicación de todos los fascículos que componen esta obra.

Distribuye para España: Marco Ibérica, Distribución de Ediciones, S.A., Carretera de Irún, km 13,350. Variante de Fuencarral, Madrid-34.
Distribuye para Argentina: Viscontea Distribuidora, S.C.A., La Rioja 1134/56, Buenos Aires.
Distribuye para Colombia: Distribuidoras Unidas, Ltda., Transversal 93, n.° 52-03, Bogotá D.E.
Distribuye para México: Distribuidora Intermex, S.A., Lucio Blanco, n.° 435, Col. San Juan Tlihuaca, Azcapotzalco, 02400, México D.F.
Distribuye para Venezuela: Distribuidora Continental, S.A., Ferrenquín a Cruz de Candelaria, 178, Caracas, y todas sus sucursales en el interior del país.

Pida a su proveedor habitual que le reserve un ejemplar de MI COMPUTER. Comprando su fascículo todas las semanas y en el mismo quiosco o librería, Vd. conseguirá un servicio más rápido, pues nos permite realizar la distribución a los puntos de venta con la mayor precisión.

Servicio de suscripciones y atrasados (sólo para España)

Las condiciones de suscripción a la obra completa (96 fascículos más las tapas, guardas y transferibles para la confección de los 8 volúmenes) son las siguientes:

a) Un pago único anticipado de 16 690 ptas. o bien 8 pagos trimestrales anticipados y consecutivos de 2 087 ptas. (sin gastos de envío).

b) Los pagos pueden hacerse efectivos mediante ingreso en la cuenta 3371872 de la Caja Postal de Ahorros y remitiendo a continuación el resguardo o su fotocopia a Distribuidora Olimpia (Paseo de Gracia, 88, 5.°, Barcelona-8), o también con talón bancario remitido a la misma dirección.

c) Se realizará un envío cada 12 semanas, compuesto de 12 fascículos y las tapas para encuadernarlos.

Los fascículos atrasados pueden adquirirse en el quiosco o librería habitual. También pueden recibirse por correo, con incremento del coste de envío, haciendo llegar su importe a Distribuidora Olimpia, en la forma establecida en el apartado b).

Para cualquier aclaración, telefonear al (93) 215 75 21.

No se efectúan envíos contra reembolso.





Programas para estudiantes

El software educativo suele basarse en el método de preguntas y respuestas, siendo las primeras muy directas y estas últimas precisas y nada ambiguas

Se espera que dentro de poco la mayor parte de las escuelas españolas puedan contar con un ordenador. No obstante, tal vez sea Gran Bretaña el país que se encuentre más avanzado en el campo de la educación asistida por ordenador (EAO) y, en consecuencia, donde la creación de software de carácter pedagógico está a la vanguardia tanto en calidad como en cantidad. En la actualidad es posible hallar en el mercado británico numerosos paquetes de repaso, que suelen exigir un mínimo esfuerzo por parte del programador. La tarea es sencilla, sólo es cuestión de presentar el texto y calificar la respuesta dada por el usuario. Una vez que se establece este procedimiento, se puede utilizar la misma estructura para dar cabida a una amplia variedad de textos relativos a diversas materias y rutinas pregunta-respuesta, con el objeto de complementar o reemplazar el libro de texto. Aunque esta clase de programas son eficaces, se debe admitir que son más bien monótonos. No obstante, no cabe duda de que las futuras generaciones de software de repaso de exámenes harán uso de las facilidades para gráficos y color de que ahora disponen la mayoría de los ordenadores personales para representar visualmente el material que ilustra el texto.

THE FRENCH MISTRESS
L = Load lesson from tape
R = Run lesson
C = Create new lesson
S = Save lesson on tape
V = Verify taped lesson
A = Add to current lesson
D = Delete entry from lesson
I = Insert new entry
T = Test lesson
40 Entries in Lesson Store

Idiomas

Para los idiomas que se suelen incluir en los planes de enseñanza existe bastante software de repaso destinado a diversas máquinas, incluyendo las ya utilizadas en las escuelas, como el BBC Micro y el ZX Spectrum, y aquellas que por lo general se utilizan más para software de juegos, como sucede en el caso de la gama Atari.

En Gran Bretaña, Kosmos Software ofrece un grupo de este tipo tanto para el Spectrum como para el BBC Modelo B en cassette; *The French mistress* (La maestra de francés), *The German master* (El maestro de alemán) y *The Spanish tutor* (El preceptor de español) son similares en cuanto a su método operativo, y los tres vienen en dos niveles. El nivel A se compone de listas de palabras o frases cortas (de hasta 59 caracteres de longitud) en inglés y en el idioma de que se trate, por categorías como "familia", "alimentos", "seres vivos" y otras similares; el cassette del nivel B abarca adjetivos y adverbios, conjugaciones y tiempos verbales.

El usuario tiene la opción de trabajar partiendo del idioma extranjero y traducirlo a su lengua, o viceversa, así como de crear listas de palabras propias, que se pueden guardar en cassette para posterior consulta. Cada uno de los cassettes está dividido en 16 lecciones, siendo la longitud máxima de cada una de ellas de 250 términos. Este límite también rige para las listas creadas por el usuario. Éste puede elegir entre la modalidad aprendizaje, en la cual se visualiza una palabra extranjera y luego su equivalente en lengua propia; un autotest, en el que sólo se visualiza uno de los idiomas, y una prueba de exactitud con tiempo medido.



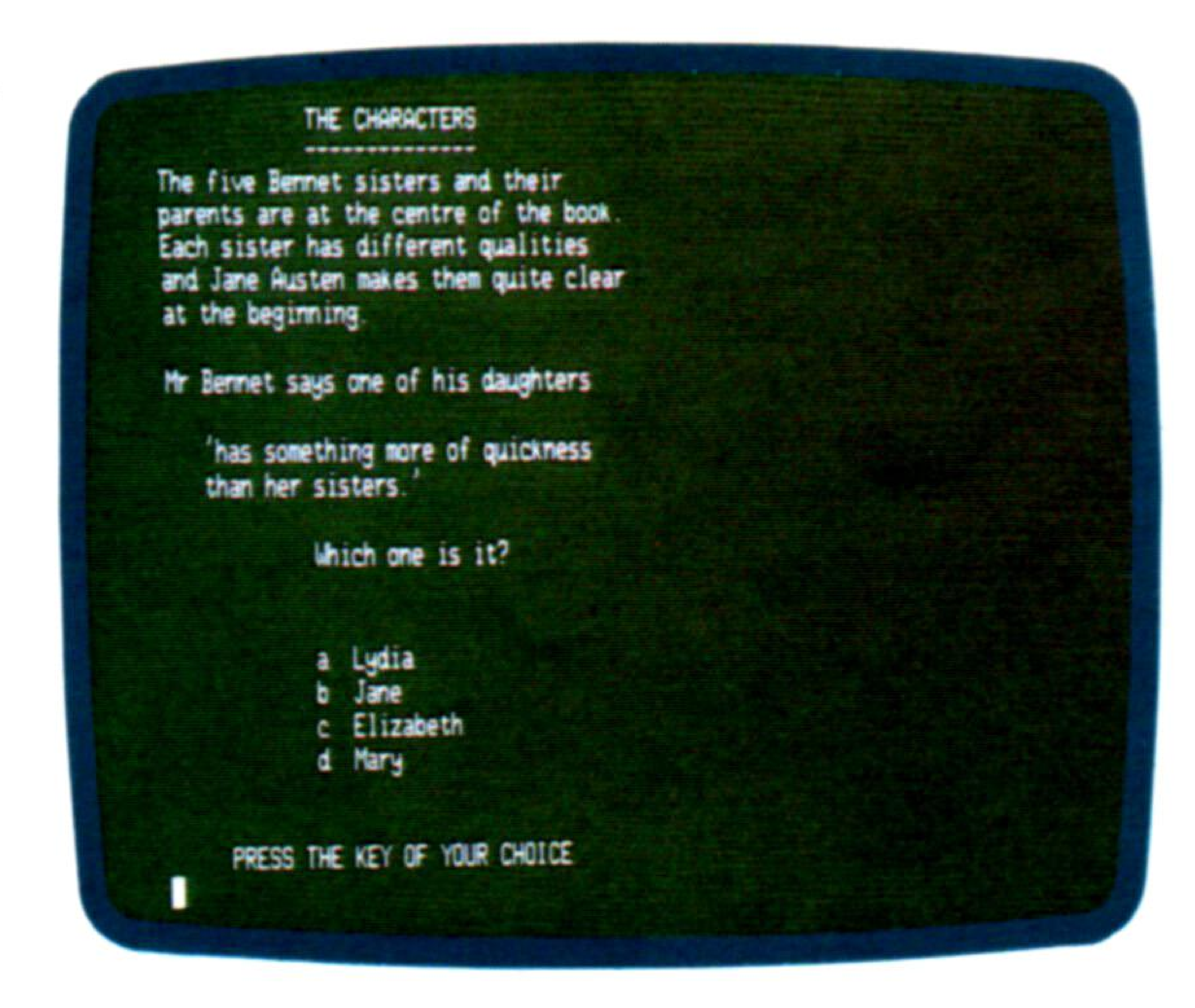
Cortesía de Pilot Software

Inglés

Los interesados en aprender inglés tienen a su disposición un variado software. *English language*

(Lengua inglesa) de Commodore, por ejemplo, está a la venta en España en versión de disco y se basa en material proporcionado por International Correspondence Schools. Forma parte de una serie más amplia para ordenadores Vic-20 ampliados. Se centra de manera primordial en la gramática y en el empleo correcto de las palabras. El menú principal ofrece elegir entre composición (conjunto de ejercicios sobre definición y utilización de las palabras, en que se formulan preguntas tipo test, cuya respuesta correcta está incluida entre otras opciones), ortografía, gramática y comprensión. El paquete se complementa con un manual que explica el método operativo mediante ejemplos adecuados y que también proporciona textos para la realización de los ejercicios de síntesis y comprensión.





Literatura inglesa

Lamentablemente para los creadores de software, los programas de estudio de literatura inglesa tanto en el nivel medio como en el avanzado se basan en obras concretas (y la lista de títulos es larga).

Pride and prejudice (Orgullo y prejuicio), la clásica novela de Jane Austen, producida por Sussex Software, constituye un excelente ejemplo del ingente esfuerzo que supone la elaboración de un paquete de este tipo. Escrito para el Research Machines 380Z (y, por consiguiente, disponible, con algunas modificaciones, para cualquier otro ordenador basado en CP/M) y la serie Commodore 3000 y 4000, el paquete se entrega en tres discos y se centra en los personajes, los temas y el argumento de la novela, así como en su contexto social.

Aunque Sussex Software se ha visto obligada a utilizar el cuestionario tipo test para evaluar el conocimiento adquirido por el alumno, cada pregunta viene reforzada por una explicación exhaustiva, y se ofrecen con la mayor profundidad posible las razones que sustentan cada respuesta.

Historia y geografía

La historia, la geografía y la economía requieren la interpretación de hechos concretos y, por lo tanto, son materias que se prestan muy bien al repaso asistido por ordenador.

En España, Commodore ha editado varios programas para facilitar el estudio de la Historia universal. En ellos se analizan la Edad Antigua y la Edad Media. En cuanto a Geografía, esta firma cuenta con programas tipo test mediante los cuales se pueden estudiar las capitales de los países asiáticos y africanos, pudiéndose modificar por el propio educador para adaptarse a otros continentes. Otro programa permite cargar mapas de diferentes zonas del mundo y enterarse dónde están localizadas las diferentes capitales. Todos estos programas están presentados en disco y en versión castellana. Atari, por su parte, dispone de programas para estudiar geografía general (dibujos de mapas del mundo y banderas de Europa).

Matemáticas y ciencias

Algunas casas distribuidoras de las grandes firmas de ordenadores personales ya han confeccionado catálogos en castellano del software disponible para el aprendizaje y la práctica de las ciencias. Así, para el Vic-20 de Commodore existe, en cinta y en disco, un programa específico de matemáticas para BUP con siete cuestionarios que cubren temas de geometría, aritmética, álgebra, etc.

Atari, por su parte, ha puesto a disposición de los alumnos de Educación General Básica y de Bachillerato dos cintas o discos para el estudio de la física y de la química. Sería interesante disponer de la biblioteca en inglés Rose Software para el Spectrum o la de SciCAL para el BBC Micro.

Son también dignos de nota los libros que comienzan a venderse en librerías con programas escritos especialmente en BASIC que dan solución a ciertos problemas y ejercicios de ciencias.



Liz Heaney



Un sistema clásico

A pesar de tener una capacidad de almacenamiento y una velocidad de operación reducidas, la tradicional unidad de disco Atari 810 cuenta con una interesante gama de órdenes

Cuando se lanzaron los ordenadores Atari 400 y 800, sin duda fueron las primeras máquinas diseñadas específicamente para su utilización personal. Casi de inmediato Atari produjo el sistema de unidad de disco simple 810 para aumentar las capacidades de los ordenadores, pero debido al elevado precio de cada unidad el sistema no consiguió hacerse popular. Aun así, con la llegada del Atari 600 compatible, es conveniente volver a analizar la unidad de disco 810.

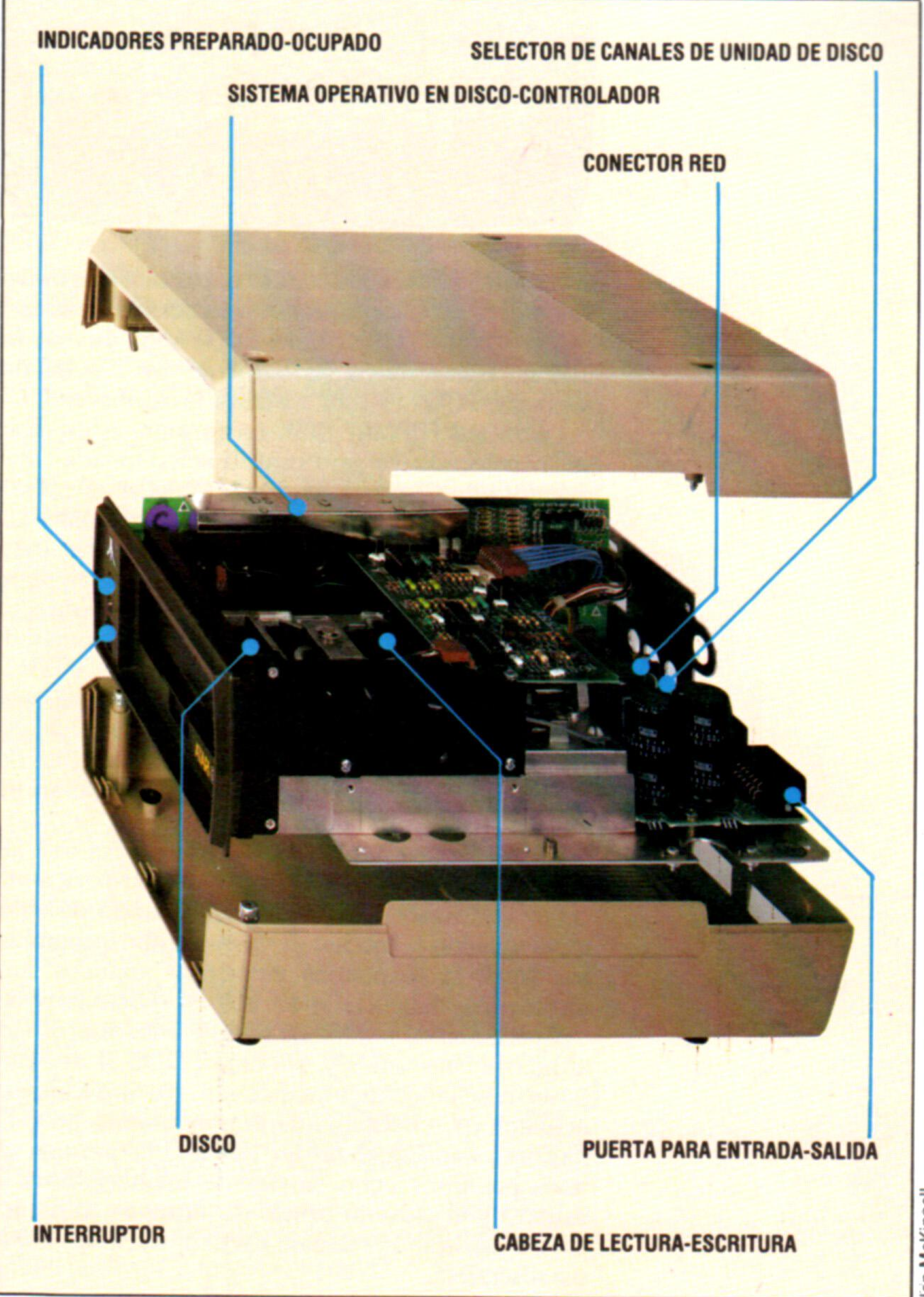
Esta unidad utiliza discos flexibles de 5 ¼ pulgadas, de densidad simple y de una sola cara. Se conecta al ordenador a través de la puerta especial de E/S en paralelo. Se pueden conectar en "cadena margarita" hasta cuatro unidades 810, numeradas de 1 a 4 conforme se establezca un interruptor de código para unidades situado en la parte posterior de cada unidad de disco. Aunque la 810 contiene su propio microprocesador, no se puede decir que sea una auténtica unidad inteligente, porque parte del DOS II de Atari (el sistema operativo en disco de que va provista normalmente) se debe cargar en RAM para poder acceder a la unidad de disco. El DOS ocupa aproximadamente cinco Kbytes de la RAM para el usuario, dejando apenas ocho Kbytes de RAM libres en aquellas máquinas de 16 Kbytes como la 600. Sin embargo, en un futuro cercano habrá tableros de RAM de 48 Kbytes para el 600, que incrementarán la RAM a 64 Kbytes.

El DOS de Atari se proporciona con la unidad de disco en un disco maestro y consta de tres archivos separados pero relacionados entre sí. Éstos son: DOS.SYS, que contiene el sistema de administración de archivos y las instrucciones de órdenes residentes en la RAM mientras se utilicen; DUP.SYS, un archivo de utilidades en disco que retiene el menú del DOS y algunas instrucciones de órdenes para el DOS; y AUTORUN.SYS, que aloja un archivo que, mediante una orden, se carga automáticamente en la RAM y se ejecuta para llamar al menú del DOS y a las porciones del DOS residentes en la RAM.

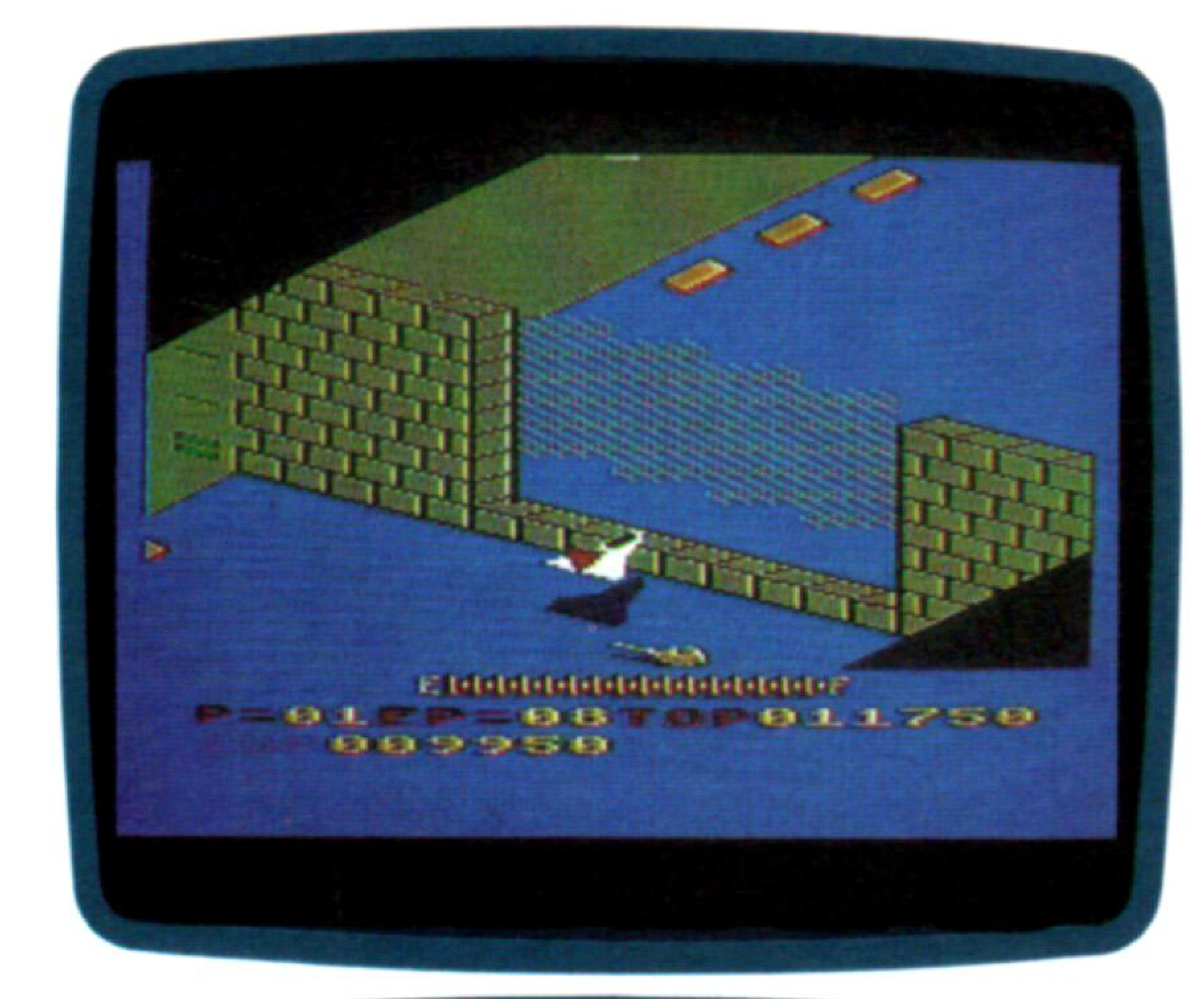
Para acceder al DOS con el cartucho de BASIC insertado, se debe encender la unidad de disco antes de darle potencia al ordenador y antes de introducir el disco maestro. Luego, al encender el ordenador (puesto en "on"), éste "calza" (o, más específicamente, carga) parte del DOS.SYX en la RAM. Para llamar al menú DOS de 15 opciones, se debe digitar DOS (RETURN).

El menú del DOS proporciona 15 opciones de órdenes para administración del disco que se seleccionan mediante las letras de la A a la O. El sistema de menú ofrece un sencillo método para administrar el disco, pero tiene una importante desventaja y es que escribe sobre una porción de la memoria para el usuario, destruyendo cualquier programa o datos que estuvieran almacenados. Esto significa que un programa que se esté editando o escribiendo se debe guardar en disco antes de llamar al menú y volver a cargarlo luego cuando se termina con el menú de opciones. El DOS posee una facilidad en virtud de la cual esto se puede realizar de forma automática si así se solicita, pero aun así esto continúa siendo incómodo.

El DOS incluye una verificación automática de los datos almacenados a cada orden de escribir, lo que limita la velocidad de transferencia de datos a



La unidad Atari 810
Los principales inconvenientes de esta unidad de disco son su interface en serie, lo que hace que la unidad funcione bastante lentamente, y su limitada capacidad de almacenamiento, 88 Kbytes por disco. No obstante, un sistema operativo en disco de relativa complejidad contribuye a atenuar estas desventajas



En el interior de la cabina
Zaxxon es sólo uno de los juegos pertenecientes a la gama que con tanto éxito ha producido Atari para sus ordenadores personales. El jugador es el piloto de un caza a reacción que, en pugna con otros aviones, efectúa un ataque a tierra. La pantalla simula el parabrisas del avión



Cortesía de Soft

Blue Max
Un segundo juego de combate entre aviones de caza producido por Atari sitúa al jugador en otros tiempos. *Blue Max*, de venta en disco, cassette o cartucho, lo sitúa a uno al mando de un avión de caza biplano de la primera guerra mundial. Aunque no pretende ser un simulador de vuelo, el punto fuerte del juego es la calidad de sus gráficos

2,4 Kbytes por segundo. Si se necesitara una mayor velocidad, cabe suprimir la verificación mediante la orden POKE 1913,80, con lo que se logra una velocidad de transferencia de datos de 4,8 Kbytes por segundo. POKE 1913,87 restaura la verificación.

La orden FORMAT DISK da formato a un disco vacío colocado en una unidad seleccionada, dividiéndolo en 40 pistas, cada una de ellas dividida a su vez en 18 sectores capaces de almacenar 128 Kbytes de datos cada uno, reservando tres bytes para el uso del sistema de administración de archivos. El DOS destina ocho sectores para el directorio del disco y un sector para la tabla del contenido de volumen (*Volume Table of Contents*: VTOC), equivalente del BAM, y reserva cuatro sectores para utilización del DOS, dando un total de 707 sectores × 125 bytes, o sea, 88 375 bytes (alrededor de 86 Kbytes) disponibles para almacenamiento.

Las órdenes estándar LOAD, SAVE y las órdenes en BASIC relacionadas se pueden utilizar para almacenar tanto archivos de programas como de datos especificando la unidad de disco de almacenamiento. También se pueden almacenar archivos cuya lectura sea secuencial o directa, a un byte por vez.

El sistema de unidad de disco 810 se diseñó hace al menos cinco años y, si bien el DOS II de Atari posee muchas configuraciones eficaces y exclusivas, actualmente el sistema ha quedado anticuado. La bajísima capacidad de 86 Kbytes, la pérdida de unos preciosos ocho Kbytes de memoria para el usuario y el elevado precio de venta se combinan para hacer que la relación calidad-precio de la 810 sea discutible.

Órdenes del disco Atari

Para la manipulación de archivos por lo general se utiliza una especificación de archivos estandarizada. Ésta es:

COMMAND "DN:NOMARCHI.EXT"

donde COMMAND es la orden DOS; N es el número de la unidad de disco (opcional 1-4 en unidad única); el NOMARCHI (nombre del archivo) contiene hasta ocho caracteres para identificar al archivo (el primero debe ser alfabético); y .EXT es el ampliador opcional de archivo (se puede emplear para indicar el tipo de datos almacenados).

Cuando se necesita esta especificación de archivo, se alude a ella como FSP. También se pueden emplear *wildcards* (tarjetas no especialmente preparadas) con algunas órdenes en las que se puede utilizar "?" para sustituir un único carácter y '*' puede reemplazar cualquier número de caracteres válidos después del '*'.

Para seleccionar una opción del menú DOS digite la letra adecuada y pulse (RETURN). En todos los casos FNM significa D1:NOMARCHI.EXT.

A. DISK DIRECTORY (directorio del disco)
(RETURN) visualiza una lista de todos los nombres de los archivos que contiene el disco de la unidad 1, el ampliador y el número de sectores de cada archivo.
B. RUN CARTRIDGE (ejecutar cartucho)
Esta opción devuelve el control del ordenador al cartucho insertado, por lo general BASIC.
C. COPY FILE (copiar archivo)
Copia un archivo de un disco a otro disco o bien al mismo bajo un nombre distinto. Por ejemplo esta línea,

D1:NOMARCHI.EXT,D2:NOMARCHI.EXT

copia NOMARCHI.EXT de la unidad de disco 1 a la 2. Del mismo modo:

D1:NOMARCHI.EXT,D1:NOMARCHI.BAK

hace una copia de seguridad de un archivo en el mismo disco. El nombre del archivo debe diferir en algo y en este caso se cambia el ampliador.
D. DELETE FILE(S) (borrar archivo-s)
FNM (RETURN) elimina uno o varios archivos especificados. Si se han de eliminar todos los archivos use '*' en lugar de nombre de archivo y ampliador.
E. RENAME FILE (cambiar nombre de archivo)
Cambia el nombre de un archivo especificado y actualiza el directorio. Por ejemplo:

D1:NOMANTI NOMACTU

Se pueden utilizar otras tarjetas *(wildcards)* para cambiar los ampliadores de un grupo de archivos.
F. LOCK FILE (proteger archivo)
FNM (RETURN) protege mediante software un archivo contra toda modificación o borrado mientras no se vuelva a dar formato al disco. Los archivos protegidos de esta manera van precedidos por '*' en el directorio visualizado.
G. UNLOCK FILE (liberar archivo)
FNM (RETURN) libera el archivo especificado protegido anteriormente o todos los nombres de archivos en cuestión si se los utiliza con otras tarjetas *wildcards*.



H. WRITE DOS FILES (escribir archivos DOS)
Siga las instrucciones visualizadas para guardar el DOS en un disco formateado.
I. FORMAT DISK (dar formato a un disco)
Siga las instrucciones para dar formato a un disco.
J. DUPLICATE DISK (duplicar disco)
Siga las instrucciones para copiar un disco entero de una unidad a otra o, copiando el contenido del disco en la memoria, en la misma unidad.
K. BINARY SAVE (guardar en binario)
FNM, CCCC, FFFF (RETURN) guarda el contenido de una zona especificada de la memoria, por lo general un programa en código de lenguaje máquina. CCCC es la dirección de comienzo y FFFF la dirección de final de un hexadecimal de cuatro dígitos.
L. BINARY LOAD (cargar en binario)
FNM (RETURN) vuelve a cargar un archivo guardado mediante BINARY SAVE en las posiciones en que estaba almacenado.
M. RUN AT ADDRESS (ejecutar en dirección)
Cuando se le solicite en la visualización, dé entrada a la dirección en la cual ejecutar un archivo cargado mediante BINARY LOAD en un hexadecimal de cuatro dígitos. (RETURN) ejecuta el programa.
N. CREATE MEM.SAV (crear MEM.SAV)
Siga las instrucciones para crear un archivo denominado MEM.SAV. Cuando se llama al menú DOS, el DOS guarda automáticamente el contenido de la memoria sobre el cual escribirá el menú y lo vuelve a cargar cuando se seleccione RUN CARTRIDGE.
O. DUPLICATE FILE (duplicar archivo)
FNM (RETURN), y a continuación siga las instrucciones necesarias para copiar archivos de un disco a otro en una única unidad. Se permiten otras tarjetas *(wildcards)*.

Las otras órdenes que controlan los archivos de programas y los archivos de datos son:

SAVE LOAD LIST ENTER RUN OPEN#
CLOSE# PRINT# INPUT# NOTE# POINT#
PUT# GET# STATUS# X10

Archivos de programas:
SAVE FSP guardar (FSP)
Escribe el programa especificado en disco en forma 'distintivada'.
LOAD FSP (cargar FSP)
Lee el programa distintivado especificado en la memoria del usuario desde la parte inferior de la memoria hacia arriba.
LIST ESP,LN1,LN2 (listar FSP,LN1,LN2)
Almacena un programa en BASIC en ATASCII (versión Atari del ASCII estándar). Especificando sólo FSP se almacena el programa entero. LN1 y LN2 representan números de línea y se pueden utilizar para especificar los números de línea inicial y final de un programa a almacenar. Se emplea conjuntamente con ENTER para mezclar programas.
ENTER FSP (dar entrada a FSP)
Lee el archivo especificado, previamente almacenado mediante LIST, en la memoria del usuario y lo mezcla con un programa ya retenido en la memoria. De existir números de línea repetidos, las líneas contenidas en el archivo recién leído ocupan el lugar de las de la memoria.
RUN FSP (ejecutar FSP)
Lee en la memoria el archivo distintivado especificado y lo ejecuta automáticamente.

Archivos de datos:
OPEN#
Esta orden controla el acceso a canales de comunicación especiales denominados *bloques de control de entrada-salida (Input/Output Control Blocks:* IOCB) y los conecta con un dispositivo apropiado, en este caso una unidad de disco y FSP:

OPEN#IOCB,AC1,AC2,FSP

donde IOCB es el canal de E/S (1-5); AC1 es el código auxiliar 1 (especifica el tipo de operación de E/S de acuerdo a una tabla que figura en el manual del DOS); y AC2 siempre es 0 para unidad de disco.

Las siguientes órdenes están relacionadas con bloques de control de E/S abiertos (OPEN) de la forma que hemos explicado arriba.

CLOSE#IOCB (cerrar IOCB)
Libera al IOCB especificado de las condiciones de E/S establecidas arriba. A los IOCB cerrados (CLOSE) no se puede acceder.
PRINT# (imprimir)
Escribe datos numéricos (X,Y) o en serie (A$) en el IOCB; y así:

PRINT#,X,Y o PRINT#,IOCB,A$

INPUT#
Lee datos numéricos o en serie del IOCB especificado; por ejemplo:

INPUT#IOCB,X,Y o INPUT#IOCB,A$

NOTE#
Se establece antes de almacenar datos mediante PRINT#. Proporciona un registro del número de sector y número de byte donde se va a almacenar el siguiente byte en disco. La lista resultante se puede almacenar como una tabla en otro archivo para el acceso a los datos al azar, de a un byte por vez si fuera necesario, mediante POINT#. Por ejemplo:

NOTE#IOCB,S,B

donde S es el número de sector (1-719) y B es el número del byte (0-124).

POINT#
Lee el byte especificado de los datos, almacenado mediante NOTE#, en la memoria para el usuario:

POINT#IOCB,S,B

PUT#
Escribe un único byte en el IOCB. Por ejemplo:

PUT#IOCB,N

donde N = de 1 a 255.

GET
Lee un único byte almacenado por PUT# con:

GET#IOCB,N

STATUS#IOCB,ERROR
Proporciona una variable especificada, en este caso ERROR, el número corriente de errores para la última operación de E/S en el IOCB. El número resultante se puede entonces verificar mediante la tabla de errores del manual del DOS de Atari.
X10 CN,#IOCB,AC1,AC2,FSB
Se utiliza para duplicar algunas de las funciones del menú del DOS mediante el empleo del número de orden relativo a la función requerida.

Precios en picado
Tomando como base 1973, el costo de un ordenador se ha reducido a la quingentésima parte por cada bit de memoria. Si un coche Rolls Royce hubiera experimentado el mismo ritmo de fabricación abaratada que el ordenador (alcanzando un nivel similar en cuanto a reducción de precio), hoy costaría 500 pesetas y su rendimiento rondaría los 300 000 km por litro de gasolina

Lógica del programador

Los operadores lógicos AND y OR son valiosas herramientas que se utilizan tanto en las instrucciones de lenguaje máquina como en la mayor parte de las versiones de BASIC

Estos dos operadores lógicos se emplean de varias formas tanto en código de lenguaje máquina como en BASIC. Un ejemplo familiar del uso de AND y OR es la relación de dos instrucciones dentro de una sentencia condicionada. Por ejemplo, diga cuál será el resultado de este programa en BASIC:

```
10 FOR I = 1 TO 5
20 FOR J = 1 TO 5
30 IF I = 3 AND J = 2 THEN PRINT I,J
40 NEXT J
50 NEXT I
60 END
```

El programa ejecutará el par de bucles anidados pero sólo imprimirá los valores de I y J si I = 3 y J = 2. Por lo tanto, este programa imprimirá en pantalla el siguiente resultado:

```
3 2
```

OR se puede utilizar de forma similar. Si corregimos la línea 30 para que se lea:

```
30 IF I = 3 AND J = 2 OR J = 4 THEN PRINT I,J
```

Se producirá la siguiente salida:

```
1 4
2 4
3 2
3 4
4 4
5 4
```

El ordenador lleva a cabo la operación AND con prioridad sobre la operación OR. I y J se imprimirán si I = 3 y J = 2, o bien siempre que J = 4. El orden de prioridad se puede modificar mediante la utilización de paréntesis. Adivine cuál será la salida del programa si se modifica así la línea 30:

```
30 IF I = 3 AND (J = 2 OR J = 4) THEN PRINT I,J
```

Aislar bits en un registro

Muchos ordenadores personales utilizan registros especiales para controlar diversas funciones de la máquina. Cada bit de un registro de este tipo puede controlar un aspecto diferente de dicha operación. Por ejemplo, en el Commodore 64 hay un registro de ocho bits que controla el encendido y apagado de los sprites. Cada uno de los bits del registro está relacionado con uno de los ocho sprites disponibles. Si alguno de los bits del registro está en 1, se hace visible en la pantalla el sprite controlado por ese bit. Si el bit está en 0, entonces el sprite se apaga y no se puede ver. Utilizando el BASIC, encender cualquier combinación de sprites es una cuestión sencilla que implica calcular el número binario de ocho bits requerido y colocar (POKE) su equivalente decimal en el registro. Sin embargo, este método no tiene en cuenta el estado del registro antes de la orden POKE y podrían apagarse sprites que previamente estuvieran encendidos. La solución del problema consiste en desarrollar una técnica que nos permita aislar los bits que se han de cambiar sin alterar ninguno de los otros.

Con el fin de ilustrar esta técnica, vamos a suponer que originalmente los sprites 0, 1, 5 y 6 están encendidos. El registro que controla la interrupción tendrá el siguiente aspecto:

Número de sprite	7	6	5	4	3	2	1	0
Bit correspondiente	0	1	1	0	0	0	1	1

Vamos ahora a encender el sprite 4, leyendo (PEEK) el registro, pero cuidando de no apagar los sprites 0,1,5 y 6 ya encendidos: enfrentaremos con un OR el byte original (01100011) y este otro (00010000), en decimal 16, que corresponde al sprite 4. Basta después colocar (POKE) el resultado:

Byte original	0	1	1	0	0	0	1	1
OR byte...	0	0	0	1	0	0	0	0
Da nuevo byte	0	1	1	1	0	0	1	1

Utilizando la orden en BASIC POKE reg, PEEK(reg) OR16 podemos ahora encender sin miedo el bit 4 en el registro. Para volver a poner en *off* el bit 4 debemos leer (PEEK) el registro y relacionar el nuevo byte (01110011) y este otro (11101111), decimal 239, con un AND:

Nuevo byte	0	1	1	1	0	0	1	1
AND	1	1	1	0	1	1	1	1
Byte original	0	1	1	0	0	0	1	1

Observe que el número 239 se puede calcular rápidamente restándole 16 a 255. Utilizando la orden en BASIC POKE reg,PEEK(reg)AND239 hemos devuelto al registro su estado original.

Estas técnicas se aplican más ampliamente en la programación en código de lenguaje máquina, donde alterar el estado de los registros de control podría constituir una parte importante del programa.

Teclado operado con la nariz

El sistema MATE (*Memory Assisted Terminal Equipment*: equipo de terminal asistido por memoria) se desarrolló en 1978 en la Universidad de Essex en un ordenador Vector 111. El programador era un espástico agudo, que manipulaba el teclado con la nariz.
MATE posee un teclado especial para producir algunos caracteres que no requiere mantener oprimida una tecla de cambio mientras se pulsa otra tecla, y almacena una base de datos de palabras a las que se puede dar entrada con la pulsación de una sola tecla

Respuestas al ejercicio 3

1a) $A.(\overline{A} + \overline{B})$
$= A.\overline{A} + A.\overline{B}.$ (propiedad distributiva)
$= A.\overline{B}$ $(A.\overline{A} = 0)$

b) $X + Y.(X + \overline{Y}) + X.(\overline{X} + Y)$
$= X + Y + X.(\overline{X} + Y)$ (relación 5)
$= X + Y + X.Y$ (relación 6)
$= X + Y$ (absorción)

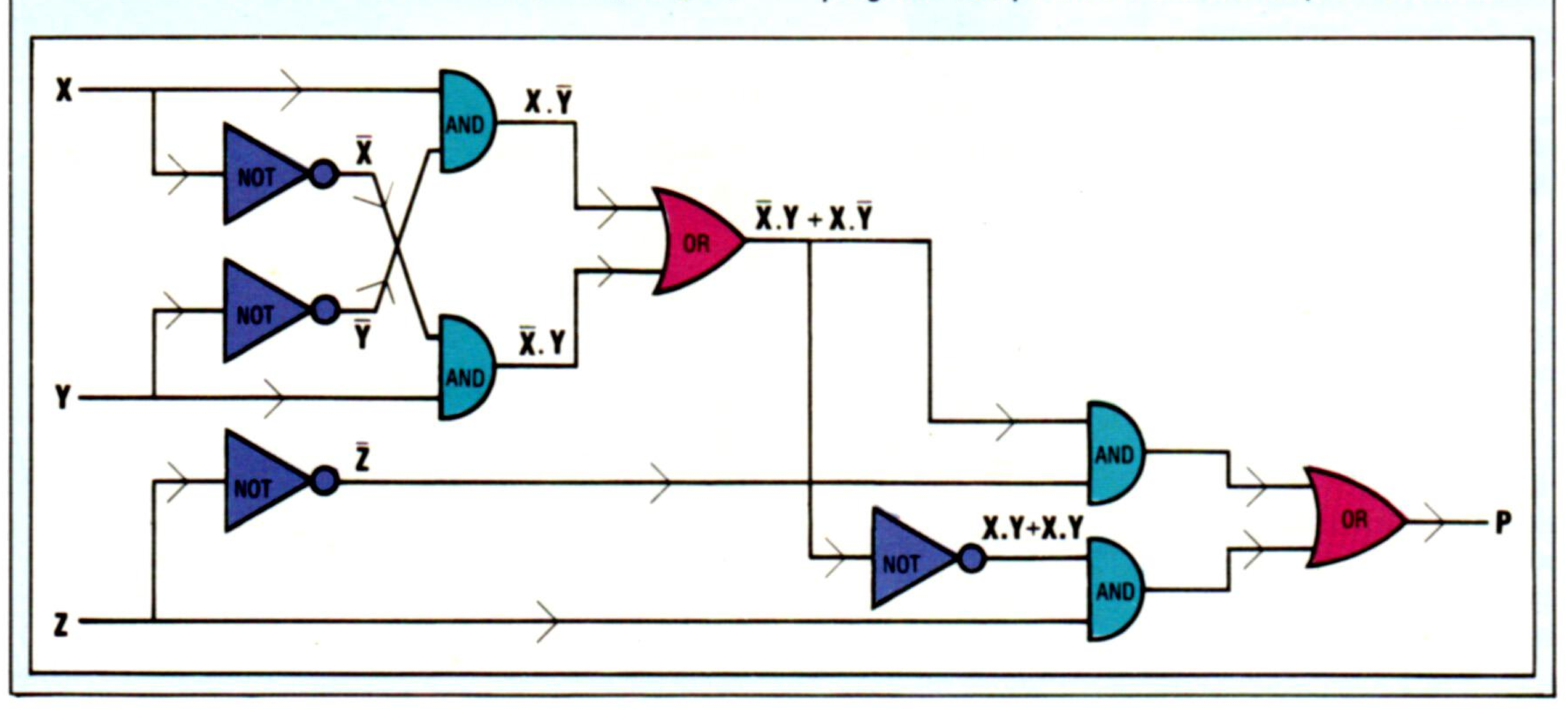
c) $P.Q + \overline{P}.Q + \overline{P}.\overline{Q}$
$= P.Q + \overline{P}.(Q + \overline{Q})$ (propiedad distributiva)
$= P.Q + \overline{P}$ $(Q + \overline{Q} = 1)$
$= \overline{P} + Q$ (dual de la rel. 6)

d) $\overline{\overline{X} + \overline{Y}.\overline{Z} + \overline{Z}.Y}$
$= \overline{\overline{X}}.\overline{Y.Z}.\overline{\overline{Z}.Y}$ (Morgan)
$= X.Y.Z.(Z + \overline{Y})$ $(\overline{\overline{X}} = X$, y además Morgan)
$= X.Y.Z.Z + X.Y.Z.\overline{Y}$ (propiedad distributiva)
$= X.Y.Z + 0$ $(Z.Z = Z, Y.\overline{Y} = 0)$
$= X.Y.Z$

3) Si los tres interruptores son X, Y y Z y la luz del recibidor es P, la tabla de verdad es:

ENTRADAS			SALIDAS
X	**Y**	**Z**	**P**
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

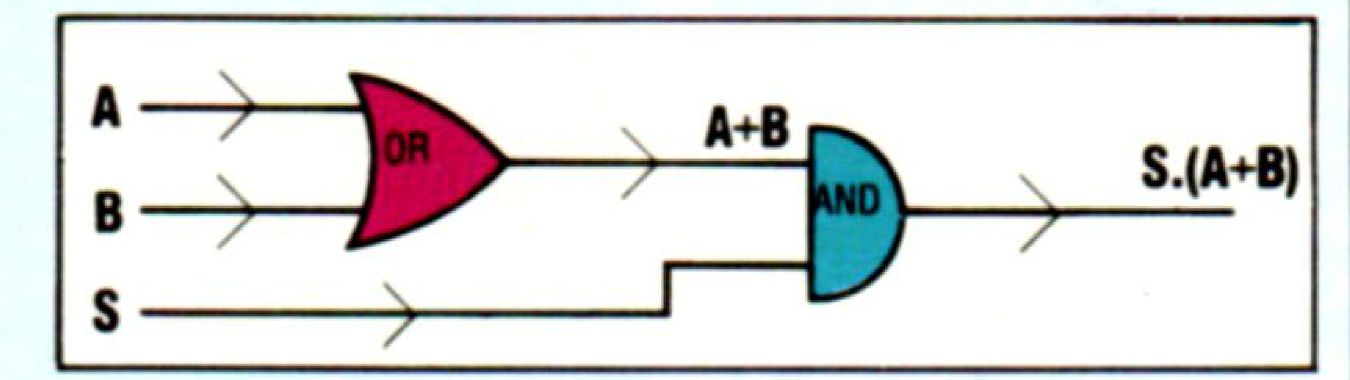
$P = \overline{X}.\overline{Y}.Z + \overline{X}.Y.\overline{Z} + X.\overline{Y}.\overline{Z} + X.Y.Z$
$= Z.(\overline{X}.\overline{Y} + X.Y) + \overline{Z}.(\overline{X}.Y + X.\overline{Y})$ (pr. dist.)
$= Z.(\overline{\overline{X}.Y + X.\overline{Y}}) + \overline{Z}.(\overline{X}.Y + X.\overline{Y})$ (Morgan)



2) La tabla de verdad para el sistema de alarma:

ENTRADAS			SALIDAS
A	**B**	**S**	**Alarma**
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Alarma $= \overline{A}.B.S + A.\overline{B}.S + A.B.S$
$= \overline{A}.B.S + A.S.(\overline{B} + B)$ (prop. distrib.)
$= \overline{A}.B.S + A.S$ $(\overline{B} + B = 1)$
$= S.(A + \overline{A}.B)$ (prop. distrib.)
$= S.(A + B)$ (dual de la rel. 6)



4) Esta tabla demuestra que la pregunta "¿Dice usted siempre la verdad?" sirve de poco, pues tanto quien suela mentir como quien diga la verdad responderán igual. La tabla posee la misma forma que la función $X.Y + \overline{X}.Y$, que se simplifica a Y. Es decir, la respuesta depende de una variable, no de dos, y la pregunta no distingue entre quienes mienten y quienes no. Ahora bien, si preguntamos "¿Tienen alas los cerdos?", la tabla es:

		POSIBLES RESPUESTAS	
		SI	**NO**
POSIBLE IDENTIDAD DE QUIEN CONTESTA	**MIENTE**	1	0
	DICE LA VERDAD	0	1

tabla de verdad para la función $X.\overline{Y} + \overline{X}.Y$, que es, asimismo, una tabla para OR exclusivo. La pregunta nos permite identificar a quien contesta.

Retirar y sustituir

Esta vez nuestro ejercicio consiste en quitar un chip de ROM y sustituirlo por un conector que acoja la memoria extraída

Si en el capítulo anterior hemos aprendido a soldar, el siguiente paso consiste en averiguar cómo deshacer conexiones soldadas, con limpieza y de manera expedita. Si sólo estamos tratando con cables y conectores, entonces no necesitamos preocuparnos mucho: probablemente sea más simple cortar el cable, tirar el enchufe y empezar otra vez con componentes nuevos. Pero ¿qué sucede si hemos de sustituir o cambiar de lugar un chip? Hasta los transistores más simples poseen tres patillas o terminales. Para liberar un transistor de su tablero impreso, necesitamos calentar las tres al mismo tiempo hasta el punto de fusión de la soldadura, de modo que se pueda quitar limpiamente el transistor; o podemos retirar las patillas de una en una. Y si la idea de liberar tres patillas a la vez es intimidante, ¿cómo serán las 40 patillas de un microprocesador mediano de ocho bits?

Sustituir un chip

Más que elegir simplemente un chip para reemplazar al azar, o agregar chips de RAM para ampliar la capacidad de una máquina, vamos a afrontar un proyecto algo ambicioso: reemplazar una ROM de BASIC del ZX81 con un conector, tirar un cable plano fuera de la carcasa hasta una placa de circuito impreso, e instalar un conector para acoger la ROM que hayamos extraído o bien cualquier otra. La razón por la que hemos elegido el ZX81 como tema para este ejercicio en particular es que el FORTH se vende como lenguaje incorporado de recambio. Además del nuevo lenguaje, la ROM también incorpora un sistema operativo multitarea que permite la operación simultánea de más de un programa, y cada uno de ellos de forma totalmente independiente respecto a los demás. En una máquina tan pequeña resulta un logro notable, aunque requiere un mínimo de dos Kbytes de RAM, lo que podría suponer la necesidad de agregar un módulo de memoria extra.

Además de ser un ejercicio en el que se pone en práctica todo lo que hemos analizado hasta el momento, un pequeño proyecto como éste también proporciona cierta experiencia en cuanto a la manipulación de componentes y le servirá como demostración de lo necesario que es ser pulcro y preciso. En el próximo capítulo de la serie analizaremos las formas de verificar el artículo ya acabado utilizando un multímetro. Este dispositivo mide corriente, voltaje y resistencia, y es una herramienta de incalculable valor para verificar y probar circuitos y componentes.

Receta

Si alguna vez ha de sustituir un chip de su ordenador personal o, como en nuestro ejemplo, debe simplemente volver a colocarlo, o si se ha de agregar un chip nuevo, es una excelente práctica instalar un conector de chip para alojarlo. Los chips con conector se pueden reemplazar en un momento; y si las sustituciones han de efectuarse con bastante frecuencia, existen a la venta refinados portachips, que minimizan el riesgo de torcer una patilla. Además de las herramientas que hemos descrito antes (véase p. 524), necesitará o bien un carrete de trenza para desoldar, o una herramienta para desoldar. El ejercicio que hemos escogido implica extraer el ROM de BASIC de un ZX81 y sustituirlo por un conector estándar desde el cual un cable plano conducirá a otro conector fuera de la máquina. El objetivo consiste en reemplazar el BASIC del ZX81 por el FORTH en ROM multitarea de David Husband, proporcionándole al mismo tiempo al usuario la posibilidad de retornar al BASIC en el futuro. En el caso de que desee usted realizar este ejercicio, también necesitará disponer de otros tres elementos: una placa de circuito impreso, un cable plano de 28 vías (ambas cosas las podrá encontrar sin dificultad en cualquier tienda de componentes electrónicos) y un trozo de poliestireno

Localizar la ROM
El primer paso consiste en abrir la carcasa y localizar la ROM de BASIC. La carcasa se mantiene unida mediante cinco tornillos, tres de los cuales están situados debajo de los "pies" de goma autoadhesivos de la superficie inferior. El relleno que no oculta ningún tornillo es el que se halla más próximo a los conectores EAR y MIC. Despegue cuidadosamente los otros tres rellenos y extraiga los tres tornillos de cabeza en cruz que hay debajo de ellos, y después los otros dos tornillos que hay a la vista. Levante la superficie inferior de la carcasa para dejar al descubierto la parte inferior del tablero de circuito impreso. Éste se puede extraer entonces de la carcasa mediante los tres tornillos de cabeza en cruz visibles, dos adyacentes al conector marginal y el otro cerca del disipador (la placa de aluminio que hay debajo del teclado). Dé vuelta al tablero. El chip de ROM que estamos buscando está situado arriba y ligeramente a la izquierda de los conectores del teclado

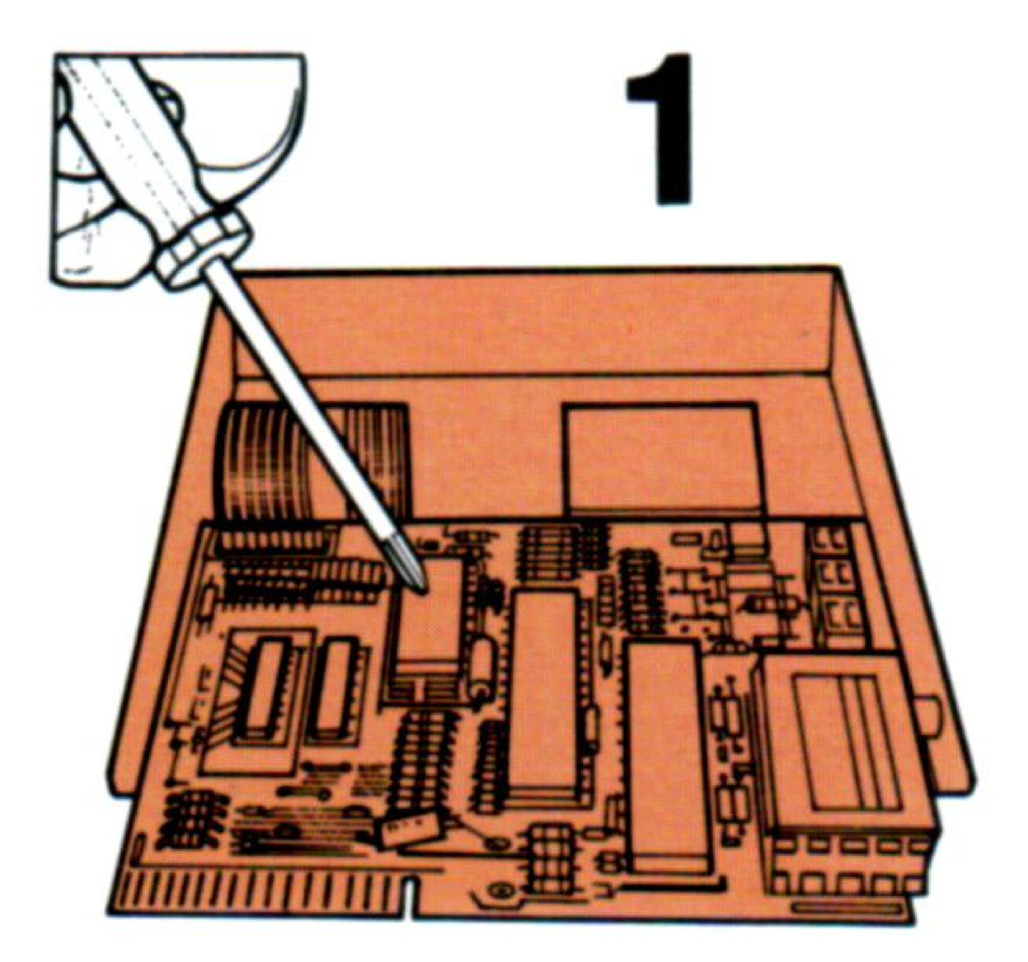
Desoldadores

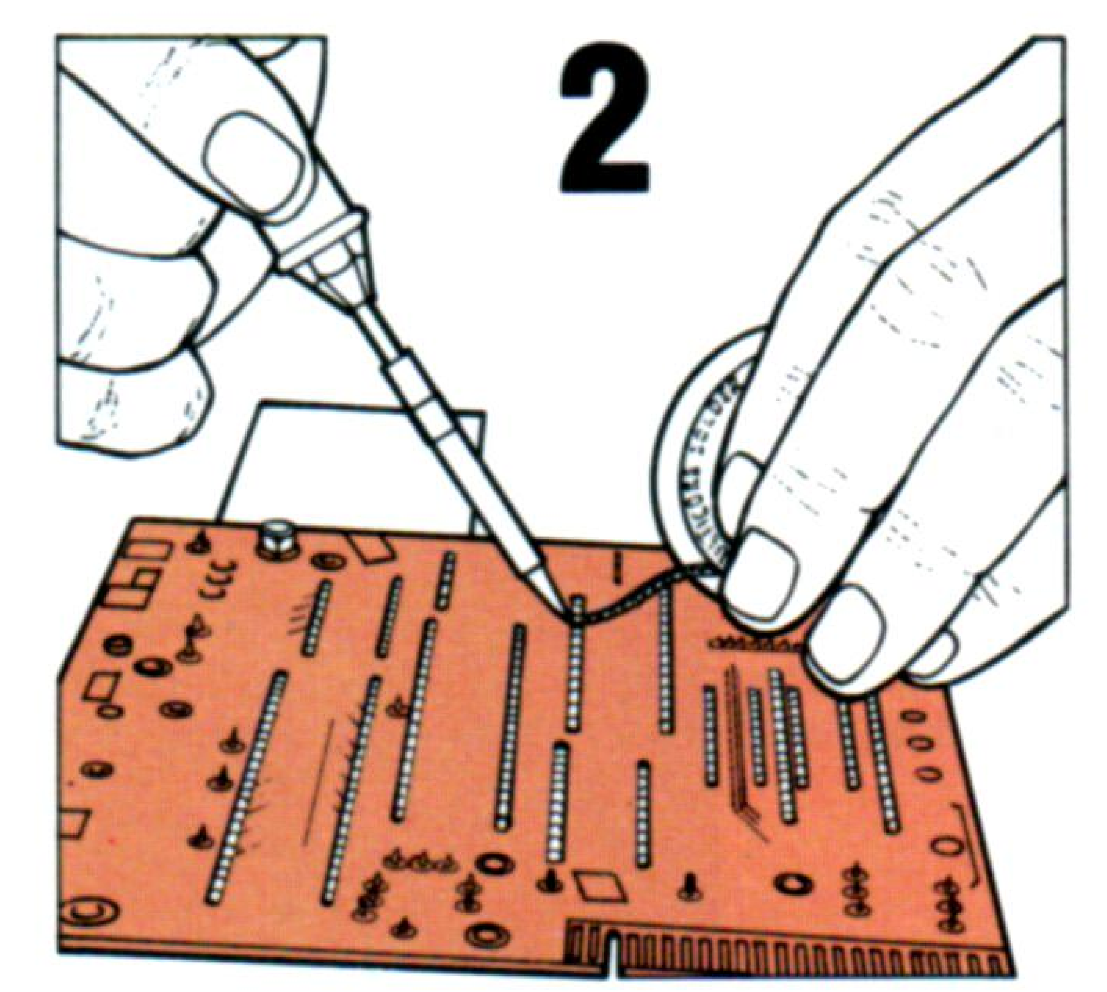
Existen dos medios patentados para desoldar. El menos caro a corto plazo es la "mecha de soldadura", alambre de cobre muy fino, trenzado en una cinta e impregnado de fundente, que se basa en el fenómeno de la acción capilar (una función de la tensión de superficie que hace que los líquidos asciendan por tubos estrechos) para absorber la soldadura fundida, así como una mecha de tejido llevaría el combustible hasta el quemador de una lámpara o de una estufa. La mecha de soldadura trenzada viene en varios anchos que están directamente relacionados con la cantidad de soldadura a eliminar. Es desechable y no se puede volver a utilizar. El segundo método es mucho más satisfactorio y consiste en un dispositivo parecido a un inflador de bicicleta con resorte, pero que funciona a la inversa: por el extremo aspira en vez de soplar. Este dispositivo tiene larga vida útil y su operación es mucho más rápida que la de las trenzas. Como puede ver, para quitar componentes con éxito es esencial seguir uno de estos dos procedimientos.



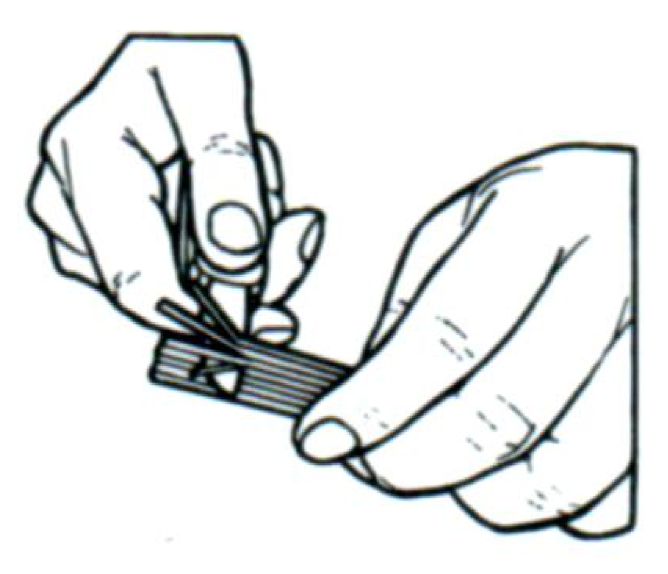
Desoldador
Caliente la juntura a desunir con el soldador hasta que corra la soldadura, aplique la herramienta para desoldar, pulse el botón disparador y la soldadura líquida será absorbida hacia el interior del cuerpo de la herramienta





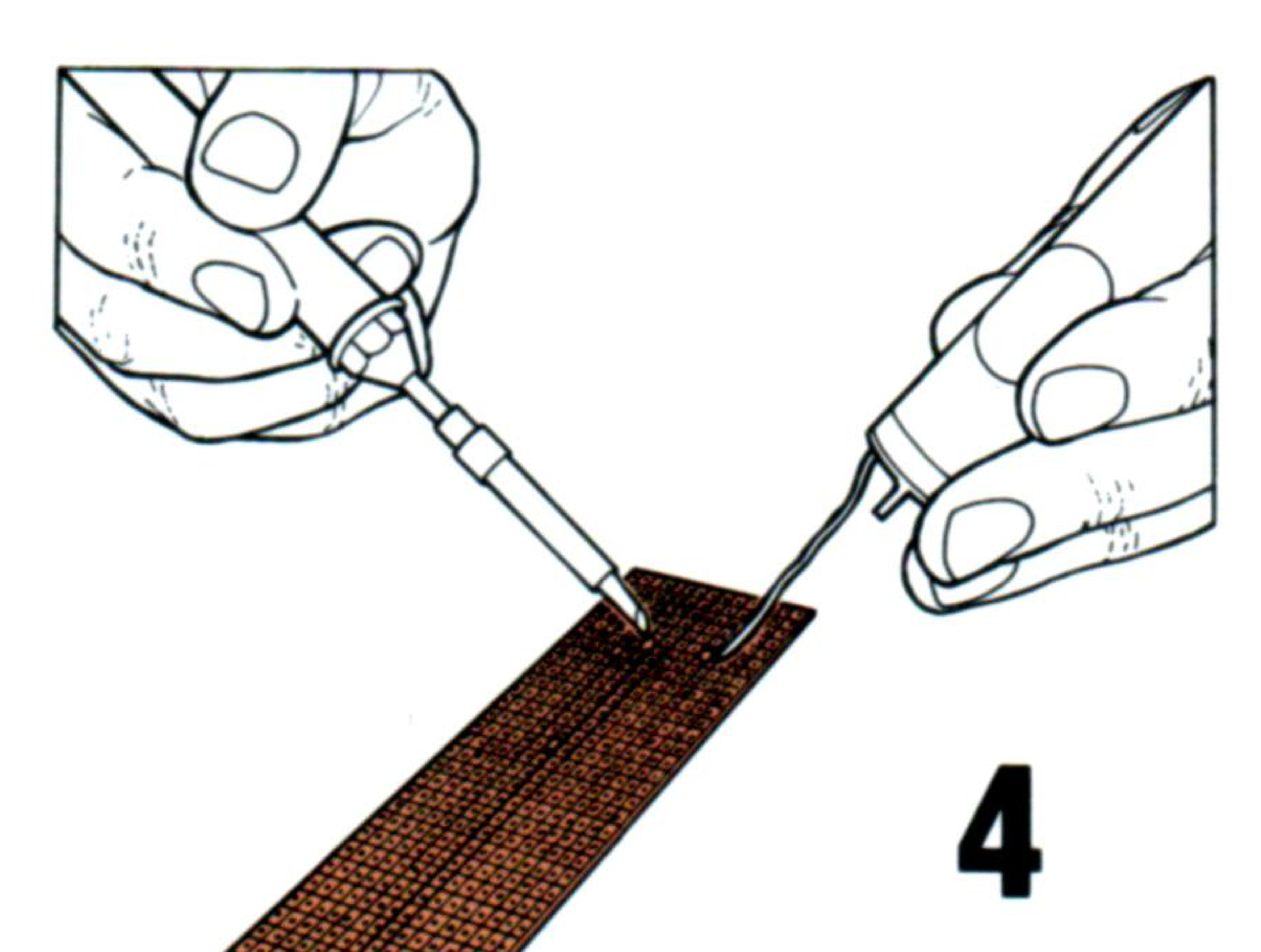


Separar la ROM
Cuando hay que desoldar, como cuando se hacen junturas, el secreto estriba en aplicar suficiente calor. Presione la trenza de soldadura sobre la juntura con la punta del soldador hasta que corra el fundente de la trenza (es probable que vea un poco de humo azul). Verá cómo la trenza absorbe la soldadura. Vaya cortando con regularidad el extremo de la trenza, ya que las secciones pequeñas se saturan rápidamente



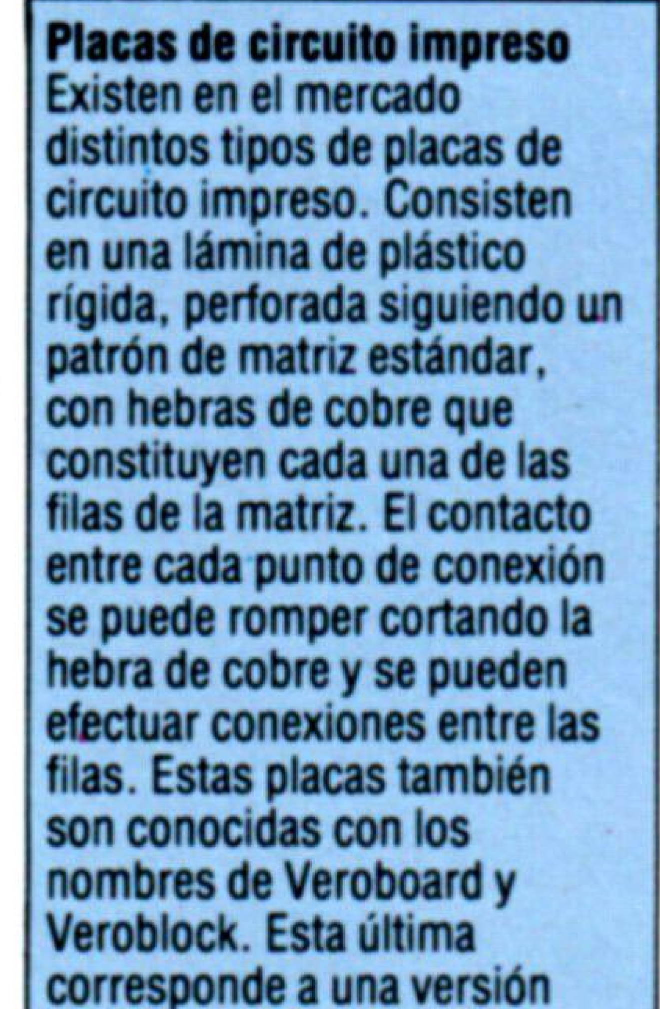
3

Cable plano
El cable plano viene en diversos anchos: nosotros necesitamos el de 28 vías (es decir, 28 cables separados) o dos tiras de 14 vías. Separe por uno de los extremos cada uno de los 28 núcleos un centímetro y dé una capa de soldadura. Trabajando desde el extremo marcado con un pequeño corte semicircular, suelde el cable plano en su sitio por ambas caras. Todos los chips poseen estos cortes, o a veces puntos, para indicar en qué forma se deben alinear

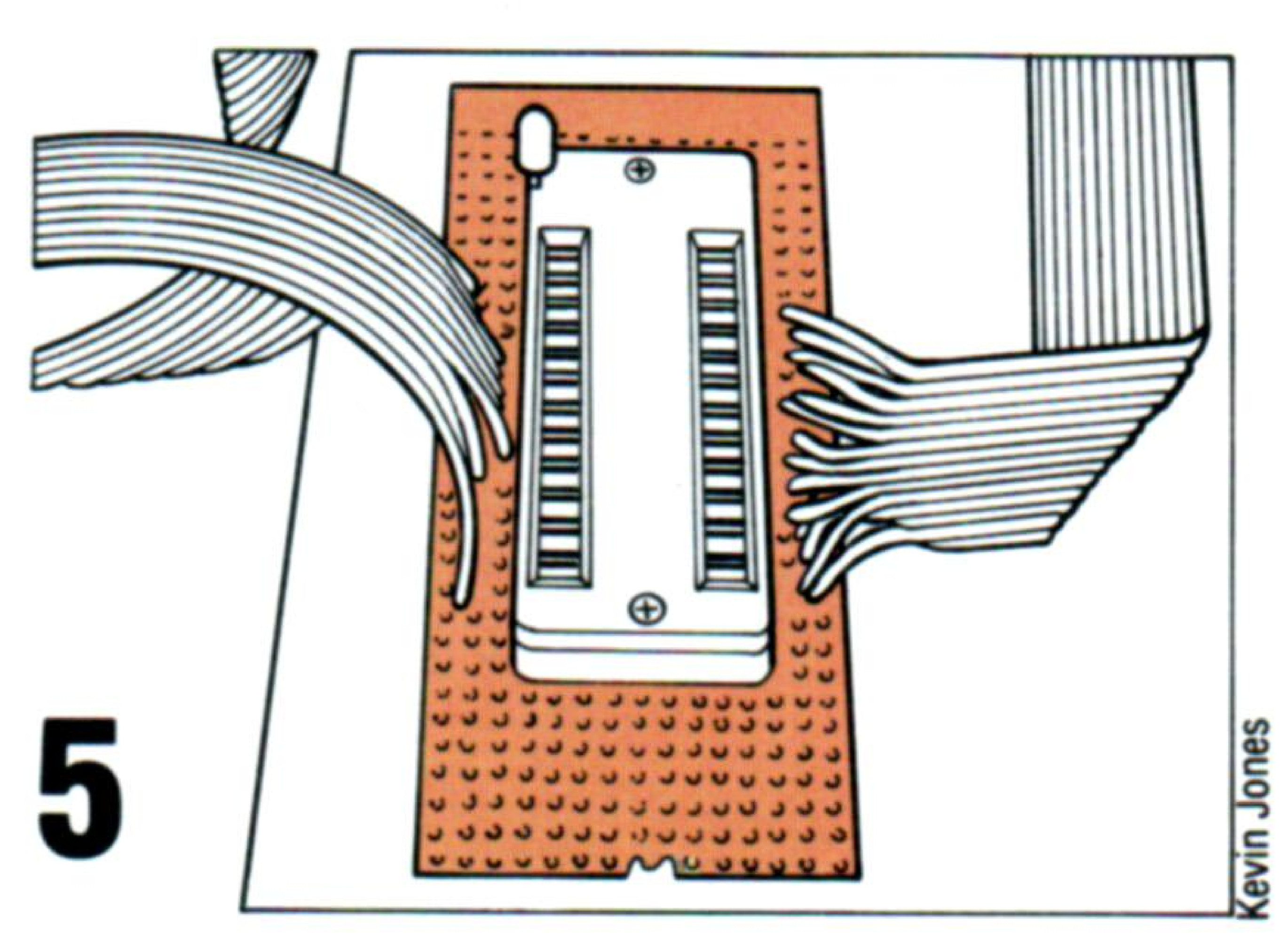


4

Conectar
Corte los dos trozos de cable plano a la longitud adecuada y pele y estañe los 28 núcleos. Luego, asegurándose de que los núcleos que parten del extremo marcado de la superficie que sostiene el chip se dirijan hacia el mismo extremo del conector, suéldelos en la placa de circuito impreso. Asegúrese de que no se cruce ninguno de los núcleos



Placas de circuito impreso
Existen en el mercado distintos tipos de placas de circuito impreso. Consisten en una lámina de plástico rígida, perforada siguiendo un patrón de matriz estándar, con hebras de cobre que constituyen cada una de las filas de la matriz. El contacto entre cada punto de conexión se puede romper cortando la hebra de cobre y se pueden efectuar conexiones entre las filas. Estas placas también son conocidas con los nombres de Veroboard y Veroblock. Esta última corresponde a una versión más sofisticada



5

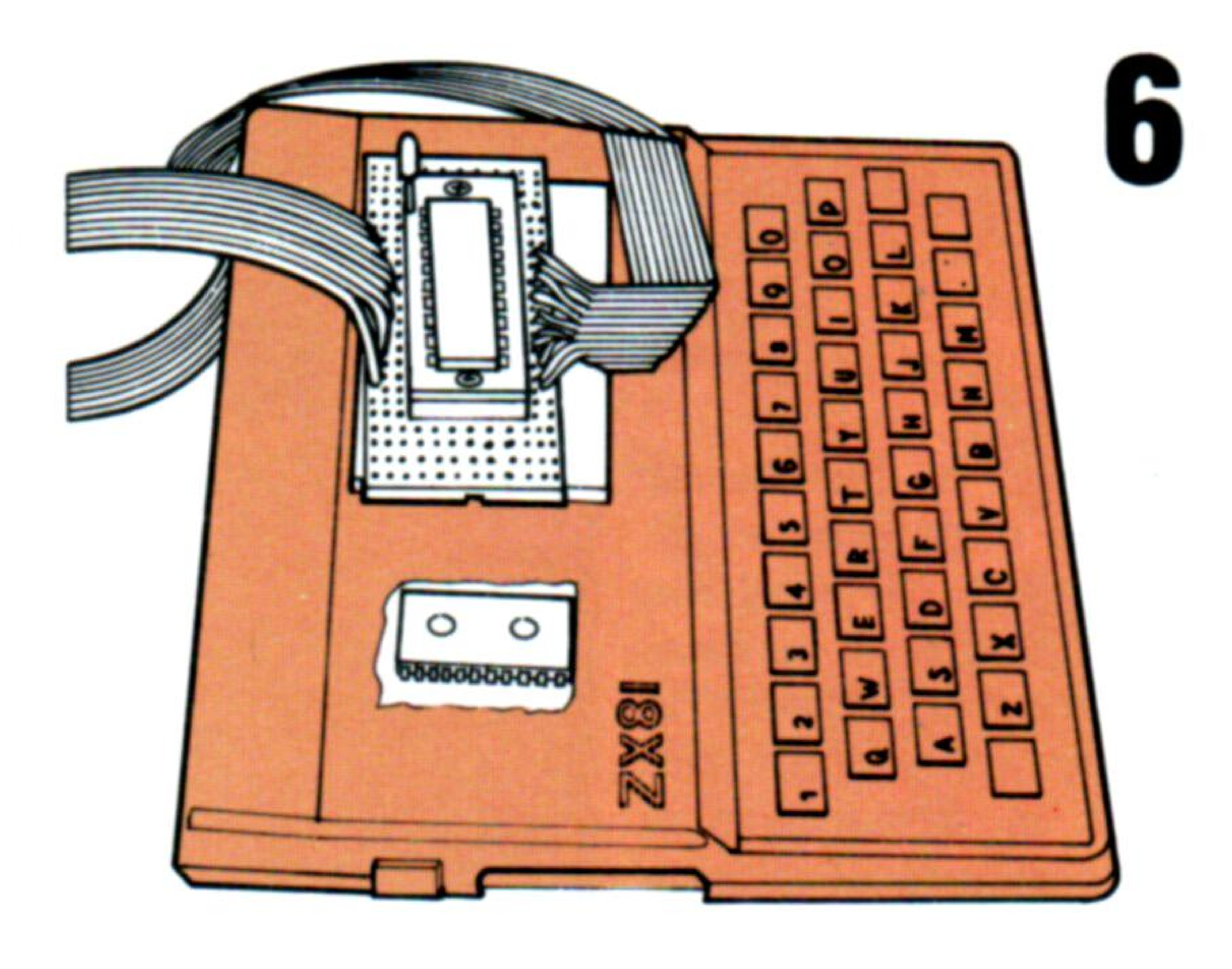
Kevin Jones

Montar el conector
Para estos fines la placa de circuito impreso es el medio más adecuado. Tiene quince puntos de contacto de ancho por aproximadamente 25 cm de largo y se puede quebrar con los dedos para obtener la longitud deseada. Coloque el conector del chip de modo que quede bien alineado respecto al aislamiento central y suelde la patilla de una esquina para sujetarlo con firmeza; luego vaya trabajando cada lado, asegurándose de que las uniones fluyan limpiamente



ATENCIÓN

La garantía de su ordenador personal (en caso de que aún estuviera en vigor) puede quedar anulada si alguien que no fuera el fabricante o un agente autorizado abriera la carcasa



6

Acabado
Cuando ya estén hechas todas las conexiones, compruebe, tanto en el propio tablero del ordenador como en el tablero hijo recién creado, que cada una de ellas esté separada de todas las demás. Cuando se trabaja con miniaturas de este tipo, en las que hay muchas conexiones unas junto a otras, éstas pueden entrar en contacto, lo que produciría cortocircuitos. Compruebe las conexiones mediante una lente de aumento y, en caso de duda, asegúrese deslizando a través del agujero una aguja u otro instrumento metálico punzante

Salida impresa

Para fines de gestión, una impresora cara, como la de rueda margarita, es casi esencial, pero para el usuario de un ordenador personal existen alternativas más económicas

La producción de una copia impresa del listado de un programa o trozo de texto está entre las prioridades del usuario de un ordenador. Lamentablemente, el costo de una buena impresora matricial podría representar el doble o el triple del precio del ordenador, mientras que un dispositivo de rueda margarita está de seguro fuera de su alcance. Pero hay buenas ofertas a precio muy económico, como las *impresoras sin impacto*.

Este nombre proviene de que no martillean agujas ni trozos moldeados de metal a través de una cinta para dejar su marca sobre un trozo de papel. Las primeras impresoras sin impacto se desarrollaron para las cajas registradoras y terminales portátiles, como la famosa serie Texas Silent 700.

Así como una impresora matricial de impacto posee una columna vertical de agujas en una cabeza que se desplaza horizontalmente a través de la página, la impresora térmica posee una columna de elementos de calentamiento. A medida que se van pidiendo puntos, el elemento correspondiente se calienta rápidamente y la diminuta superficie de papel debajo del elemento cambia de color.

Aparte de las terminales de Texas Instruments, las impresoras térmicas también fueron adoptadas por Apple, que produjo una pequeña impresora para su ordenador denominada *Silentype*, y Mattel, que ofrece una unidad similar para utilizar con su ordenador personal Aquarius.

Por su parte, el mecanismo que emplea la impresora electrostática es moderadamente ruidoso y, salvo un ejemplo temprano de Centronics denominado Microprinter P1, no consiguió la aceptación general hasta que sir Clive Sinclair adoptó el sistema para la impresora ZX Printer, de la cual se han vendido enormes cantidades como periférico exclusivo para el ZX81 y el ZX Spectrum.

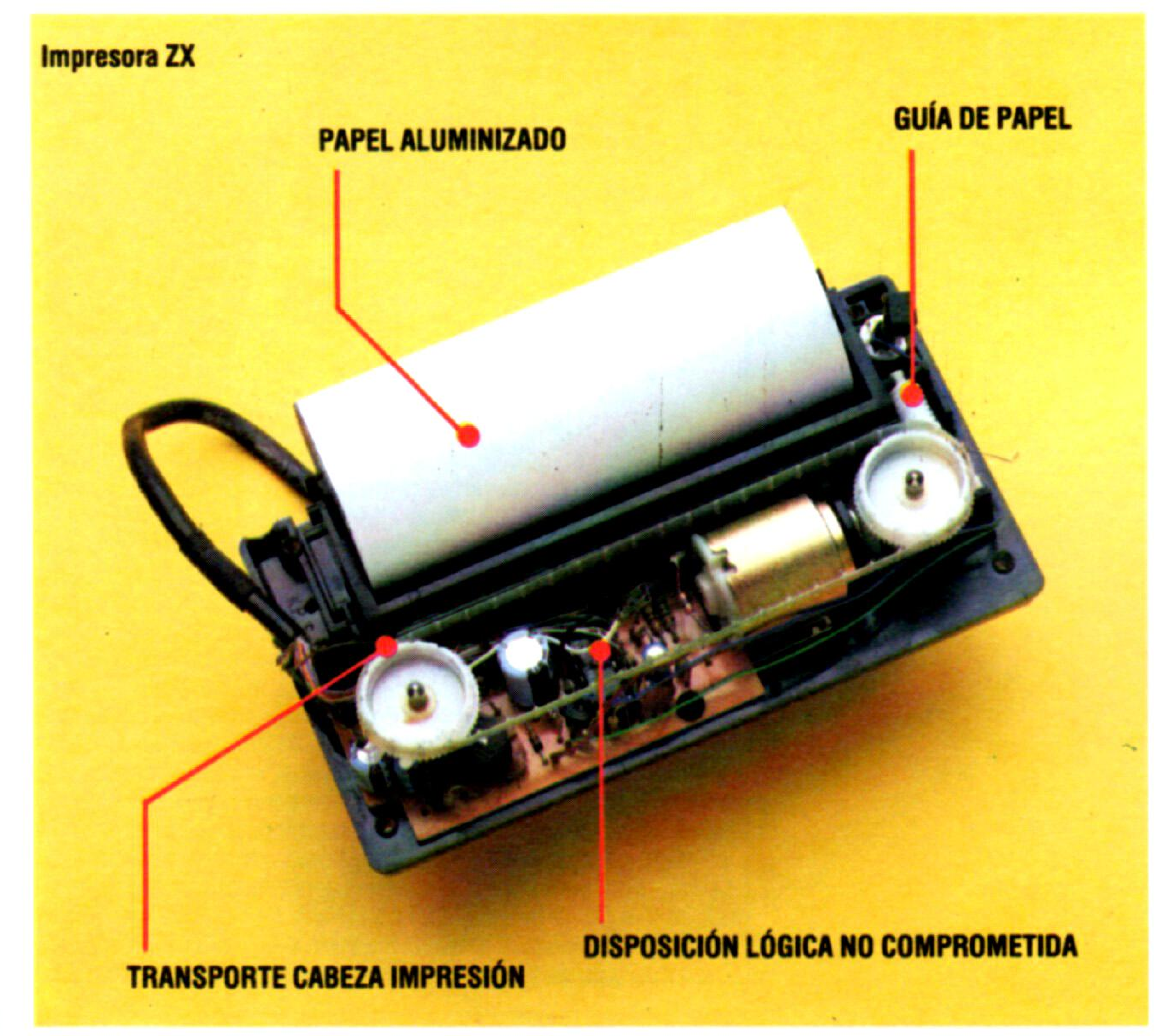
El principio en el que se basan las impresoras electrostáticas es una cabeza de impresión de un único cable que se arrastra a través de un papel con un revestimiento especial. En cada uno de los puntos que se requieren para construir un carácter, la impresora genera una chispa que quema el delgado revestimiento metálico y deja al descubierto el papel negro de fondo. Sinclair mejoró el sistema utilizando dos cabezas en un cinturón continuo,

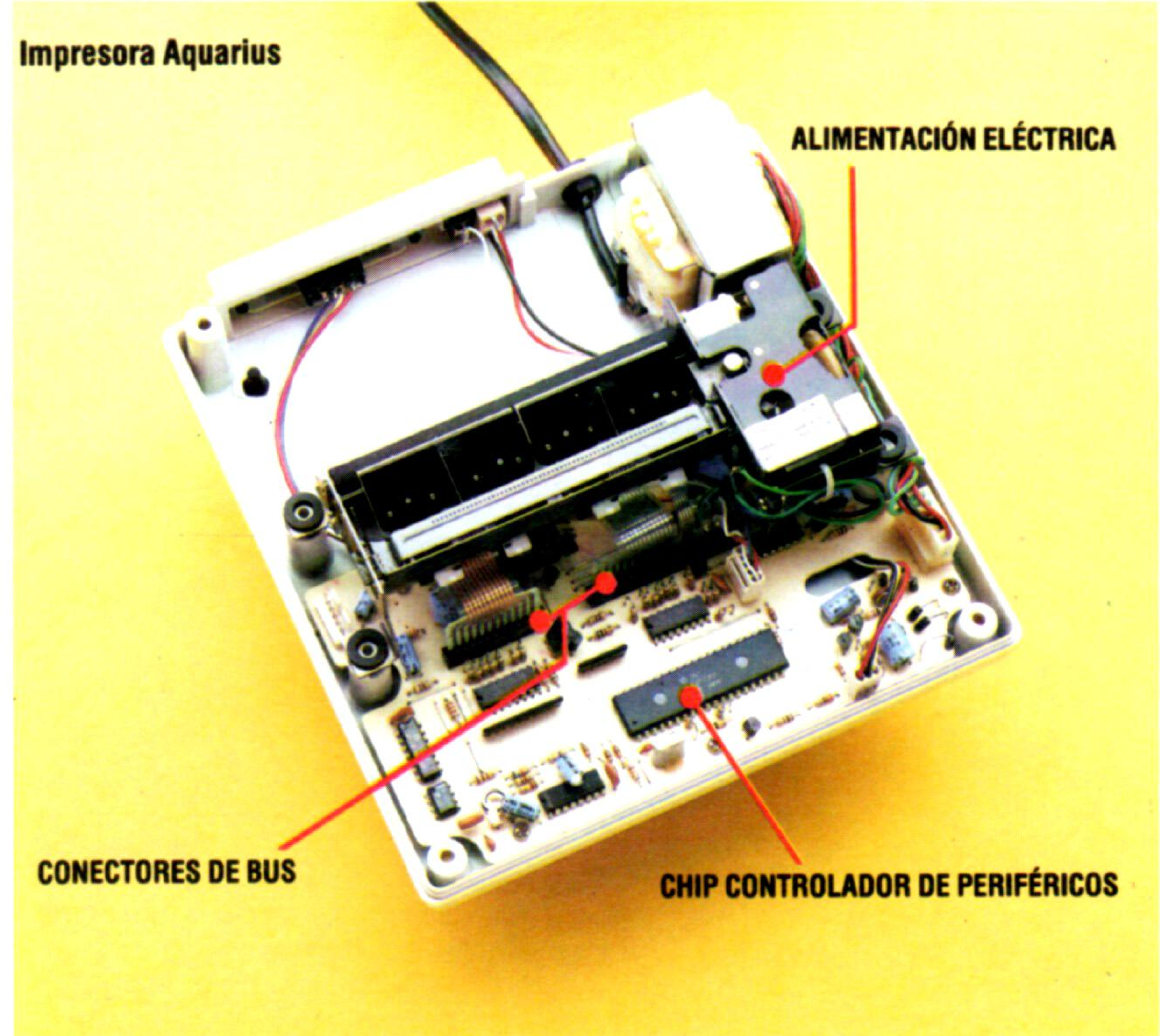




La impresora ZX
Introducida como dispositivo de salida para el ZX81, la impresora ZX también es compatible con el Spectrum. Ofrece un método económico en virtud del cual los usuarios de ordenadores personales pueden obtener resultados y listados de programas impresos. Aunque su reducido costo resulta interesante, su dependencia del papel electrostático la deja en clara desventaja

La impresora Aquarius
Para su microordenador personal, Aquarius optó por el otro procedimiento de bajo precio: la impresión térmica (aunque también hay a la venta una impresora económica a cuatro colores que utiliza lápices esferográficos). Al igual que la impresora ZX, adolece del inconveniente de que depende de un papel especial





Chris Stevens

pero todavía siguen siendo necesarias ocho pasadas de una cabeza para crear cada fila de caracteres. Por suerte, la impresora ZX sólo tiene que imprimir 32 caracteres en cada línea, de modo que la velocidad es aceptable.



Unidad con clase
De todas las representantes de la tecnología de impresión térmicas, pocas son tan elegantes como la Brother EP-22. Por menos de lo que valen muchas de las impresoras más baratas de cualquier tipo, usted no sólo adquiere una impresora térmica de 75 columnas sino, por añadidura, una máquina de escribir portátil. Diseñada para operar con papel térmico y como una impresora normal de impacto con una cinta acoplada, la unidad puede almacenar hasta 2 000 caracteres en su memoria, permitiendo crear cartas o recordatorios sobre la marcha. La memoria electrónica en realidad no admite el tratamiento de textos como tal, pero se puede corregir cualquier carácter de los últimos 16 a los que ha dado entrada, ya que, antes de impresionar el papel, la salida va a una visualización en cristal líquido incorporada. La calidad de la salida no es equiparable con la de una auténtica "impresora" (las letras en minúscula carecen, por ejemplo, de auténticos trazos bajos), pero para acompañar ordenadores tales como el Epson HX-20 y el Tandy Modelo 100 cumple una función que sería difícil de obtener con otra forma de impresora de impacto

El principal inconveniente de ambos tipos de impresora es que utilizan papel especial. Los materiales de este tipo suelen ser caros y sólo se venden en rollos, lo que dificulta el almacenamiento. En el caso de la impresora térmica, es esencial adquirir un papel de la calidad adecuada; de lo contrario, la imagen no se desarrollará correctamente, además de perder intensidad con el tiempo o la exposición al calor. El papel electrostático es aún más delicado y, si es manipulado con las manos húmedas o sudorosas, la imagen se torna confusa y se va diluyendo, ya que el revestimiento se disuelve. En ambos casos la forma de asegurar una imagen buena y duradera consiste en hacer una fotocopia.

A pesar de tales inconvenientes, estos dos tipos de impresora ofrecen a los fabricantes de los ordenadores personales más pequeños la posibilidad de proporcionar alguna forma de sistema de impresión para sus máquinas, a la que, en otras circunstancias, tendrían que renunciar. El método de impresión matricial permite realizar copias directas de la visualización en pantalla, de modo que se pueden reproducir tanto textos como gráficos sin costo adicional, si al usuario no le preocupa demasiado la relativamente baja calidad de la imagen final.

Recientemente, sin embargo, el dominio que ejercen estos sistemas de impresión ha sido desafiado por los plotters de lápices de cuatro colores que suministran Tandy y Sharp. Tales lápices pueden producir juegos de caracteres excelentes, totalmente formados, además de soberbios gráficos de líneas en rollos de papel normal, por aproximadamente el mismo costo que los dispositivos electrostáticos o térmicos. A medida que se va desarrollando la tecnología de impresión, el dispositivo sin impacto, con su sencillo mecanismo y sus limitadas capacidades, sigue siendo el que mejor satisface las necesidades de impresión del mercado de ordenadores personales económicos.



La impresora térmica de Tandy
Más adaptable que las otras dos impresoras de precio económico que hemos descrito, la Tandy TP-10 es compatible con todos los microordenadores de la gama Tandy

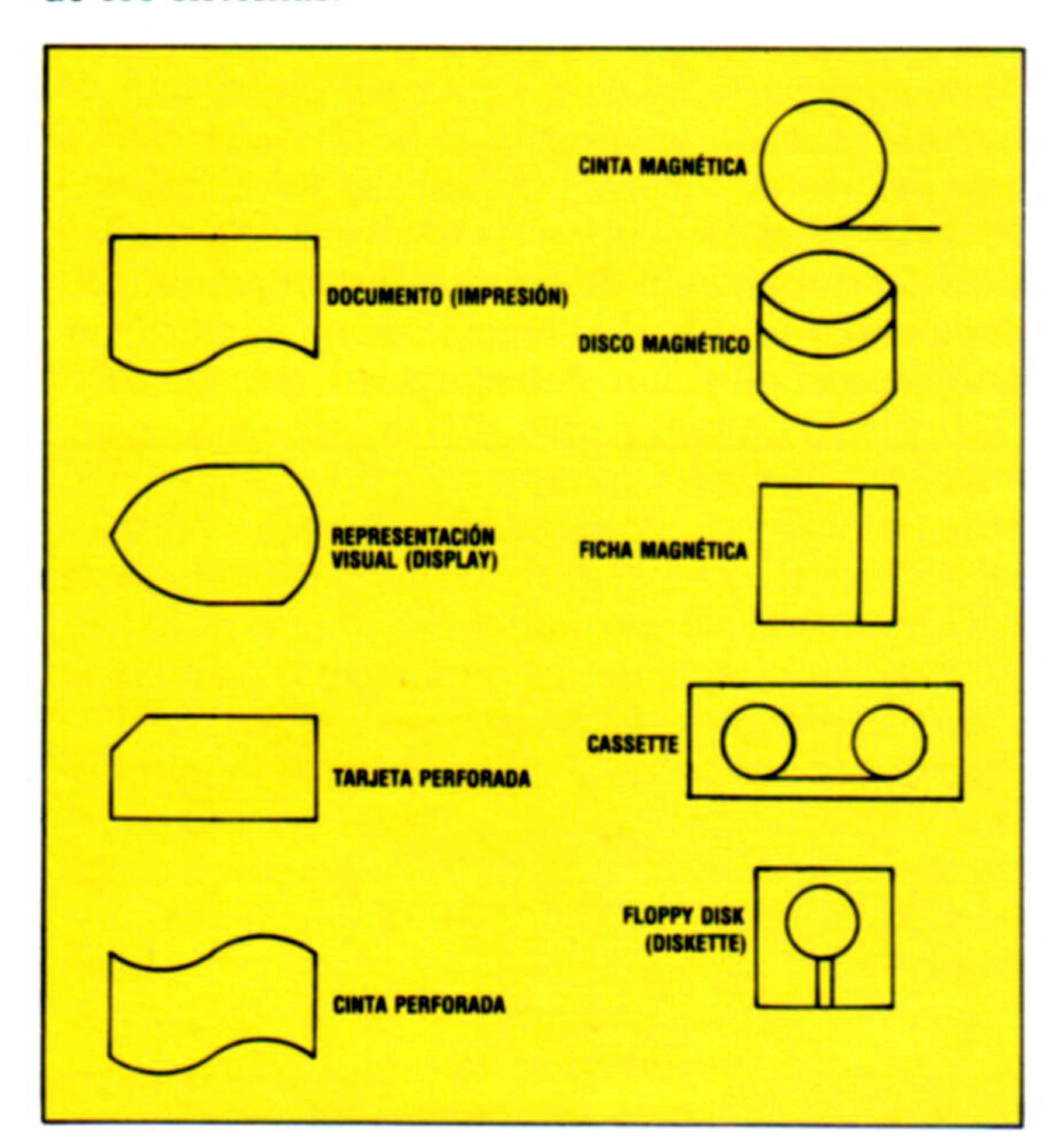
Ian McKinnell

Simbología

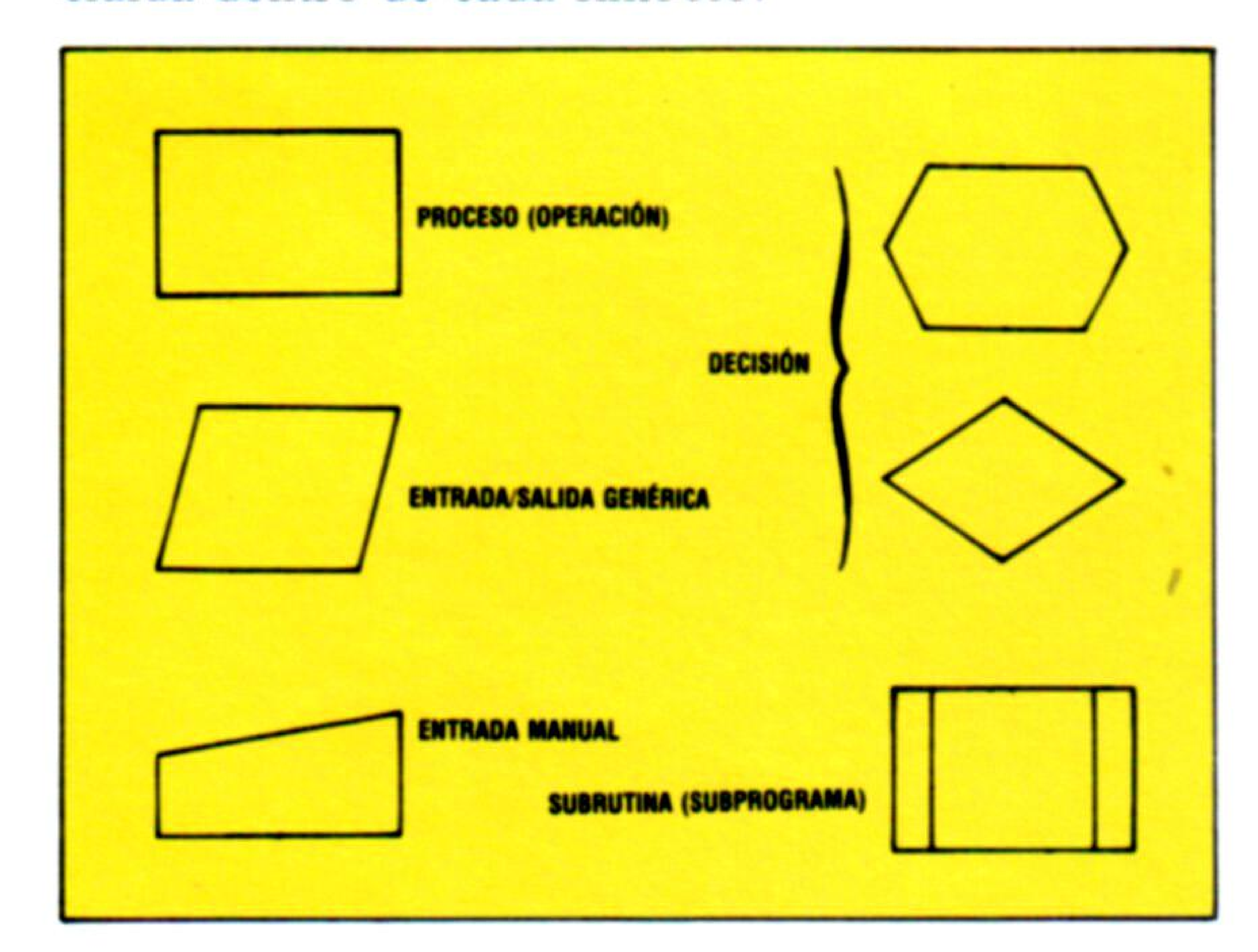
Hemos reunido en esta página los dibujos geométricos que nos servirán de símbolos en posteriores diagramas de flujo, a medida que éstos vayan siendo más complejos

Los diagramas se componen, tal como se ha visto, de una sucesión de símbolos gráficos que representan de forma individual las operaciones que realiza el ordenador. Dichos símbolos pueden clasificarse en cuatro apartados: de sistema, de proceso, auxiliares y de líneas de flujo.

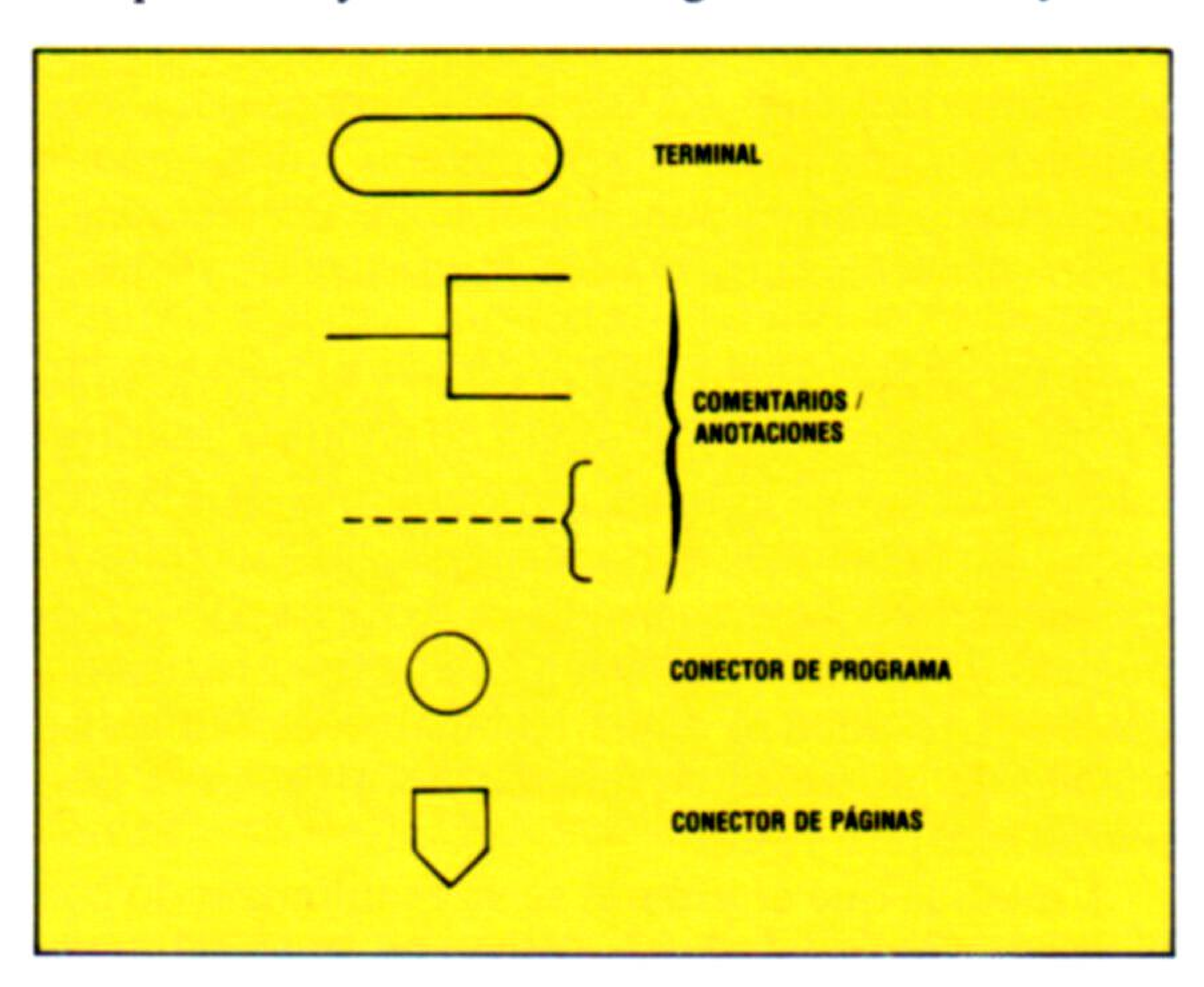
Símbolos de sistema: Son los que representan soportes de datos, ya sean manuales o automáticos. Representan asimismo las unidades y componentes de los sistemas.



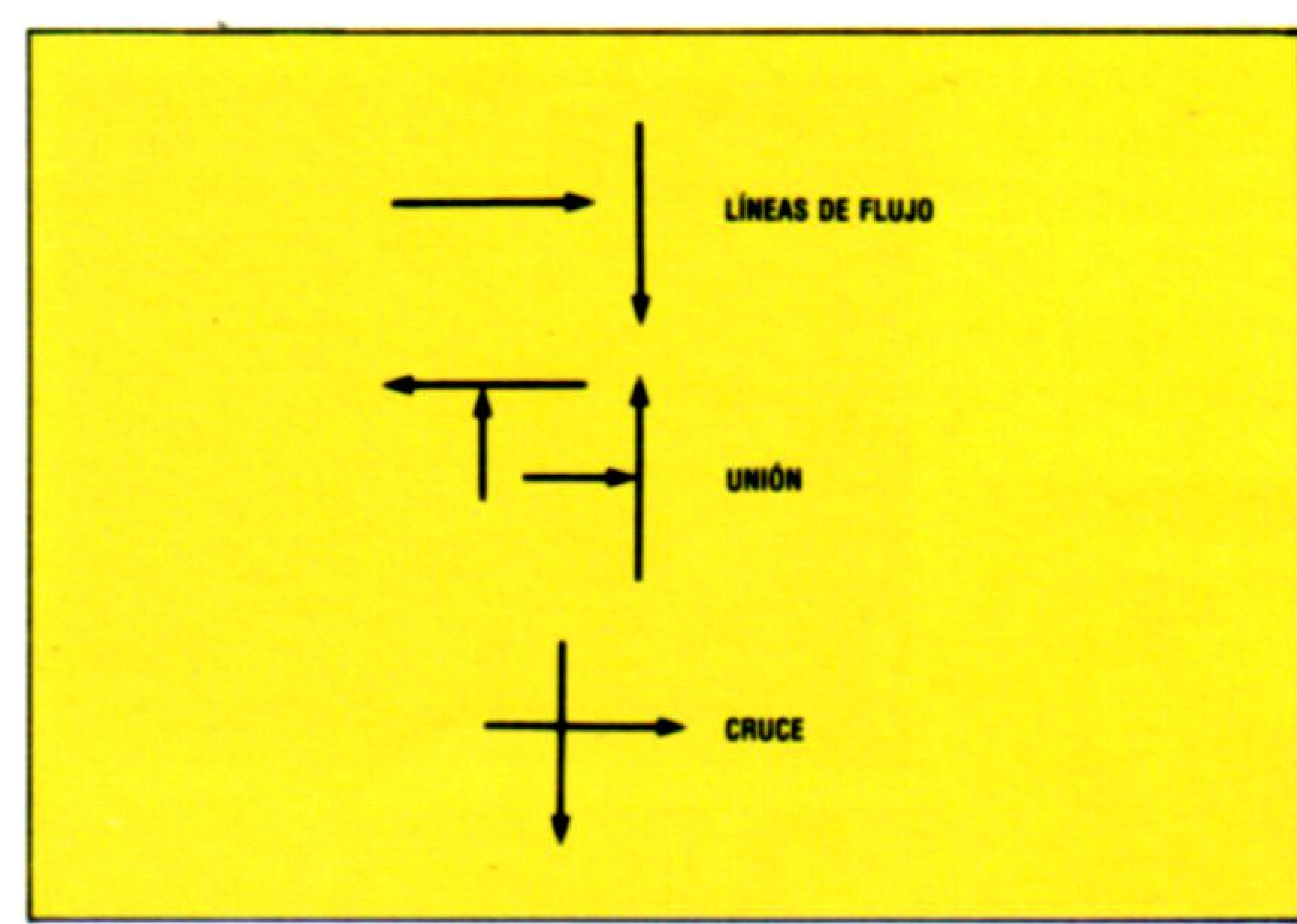
Símbolos de proceso: Representan el desarrollo de las operaciones individuales según la expresión incluida dentro de cada símbolo.



Símbolos auxiliares: Se utilizan para dar mayor comprensión y claridad al diagrama en su conjunto.



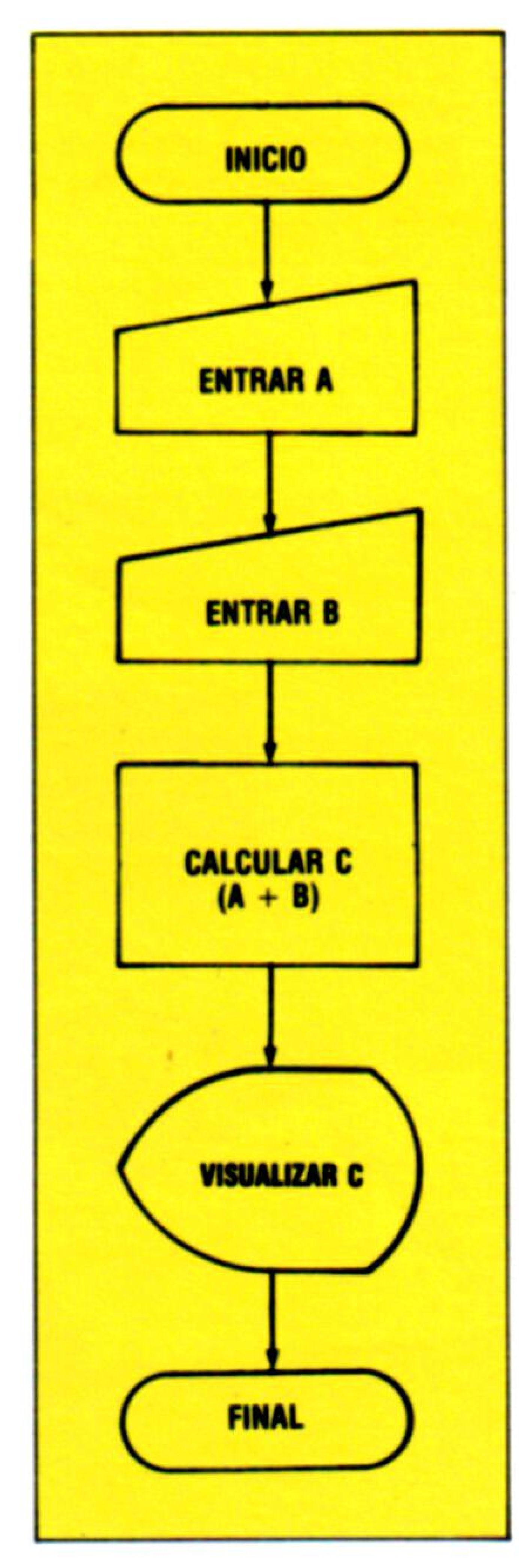
Líneas de flujo: Se las podría considerar como un subapartado del anterior grupo; son las que marcan el orden de la secuencia de operaciones y que envían hacia otros puntos del diagrama, conectándose entre sí.



Paulatinamente, y conforme aparezcan en los problemas que se vayan realizando, se explicarán éstos y otros posibles símbolos o combinaciones, así como la función que cumplen dentro del diagrama.

El ejemplo de la derecha muestra cómo, mediante la introducción de dos valores por teclado, se llega a la visualización del valor resultante tras la operación de la suma de dichas cantidades. Sirve, además, para comentar el símbolo perteneciente al grupo de auxiliares conocido por *terminal*. Dicho símbolo muestra indistintamente tanto el inicio (START) como el final (END) del organigrama.

Su utilización es obligatoria en todo diagrama.



Rapidez y eficiencia

Esta vez analizaremos dos expeditos y recomendables paquetes de tratamiento de textos creados para el BBC Micro

Uno de los paquetes más populares de tratamiento de textos para el BBC Micro es Wordwise, que está almacenado en un chip que se acopla en un conector para ROM del ordenador. Una vez acoplado, digite *W y el BBC se convertirá en un procesador de textos. El paquete Wordwise, tal como se entrega, consta del chip ROM, instrucciones para acoplamiento, un manual de 30 páginas y una cassette que visualiza en la pantalla el texto del manual e ilustra provechosamente las funciones de diversas órdenes.

El Wordwise es de fácil manejo y se puede utilizar de inmediato sin necesidad de adquirir adiestramiento previo y sin tener que remitirse de manera constante al manual. Para iniciarse, tan sólo elija la modalidad edición pulsando la tecla Escape y comience a digitar. La edición la efectúa el cursor y las letras se pueden digitar o borrar. El Wordwise se vale de las teclas de función del BBC para desplazar y copiar texto.

A medida que el usuario se va familiarizando con la mayor flexibilidad que ofrece un procesador de textos en comparación con la máquina de escribir o el lápiz y el papel, puede ir explorando con mayor profundidad las configuraciones. Enseguida entran a formar parte del repertorio del usuario el saltar hasta el principio o el final de una línea o página utilizando las teclas Shift y Cursor, o sobreescribir letras mediante el empleo de una tecla de función.

Los caracteres aparecen en la pantalla en la modalidad de 40 columnas y son totalmente legibles en un televisor doméstico. El aspecto final del texto se puede ver previamente en la pantalla en la modalidad de 80 columnas mediante una tecla de función. Ésta permite apreciar cómo aparecerá el texto una vez impreso. Es probable que en esta etapa de visión previa el recién iniciado en el Wordwise considere primero el trazado general y el aspecto del texto y necesite remitirse a la sección del manual que incluye las órdenes definibles por el usuario. Estas órdenes se ocupan de la faceta de presentación del tratamiento de textos. Como ejemplo, podemos citar la orden LM seguida de un número, que establece la posición del margen izquierdo (*Left*

El equipo completo

Una instalación para tratamiento de textos debe incluir teclado y monitor (o televisor), disco y unidad de disco para almacenamiento, impresora y útiles de escritorio para una salida impresa

MONITOR

IMPRESORA

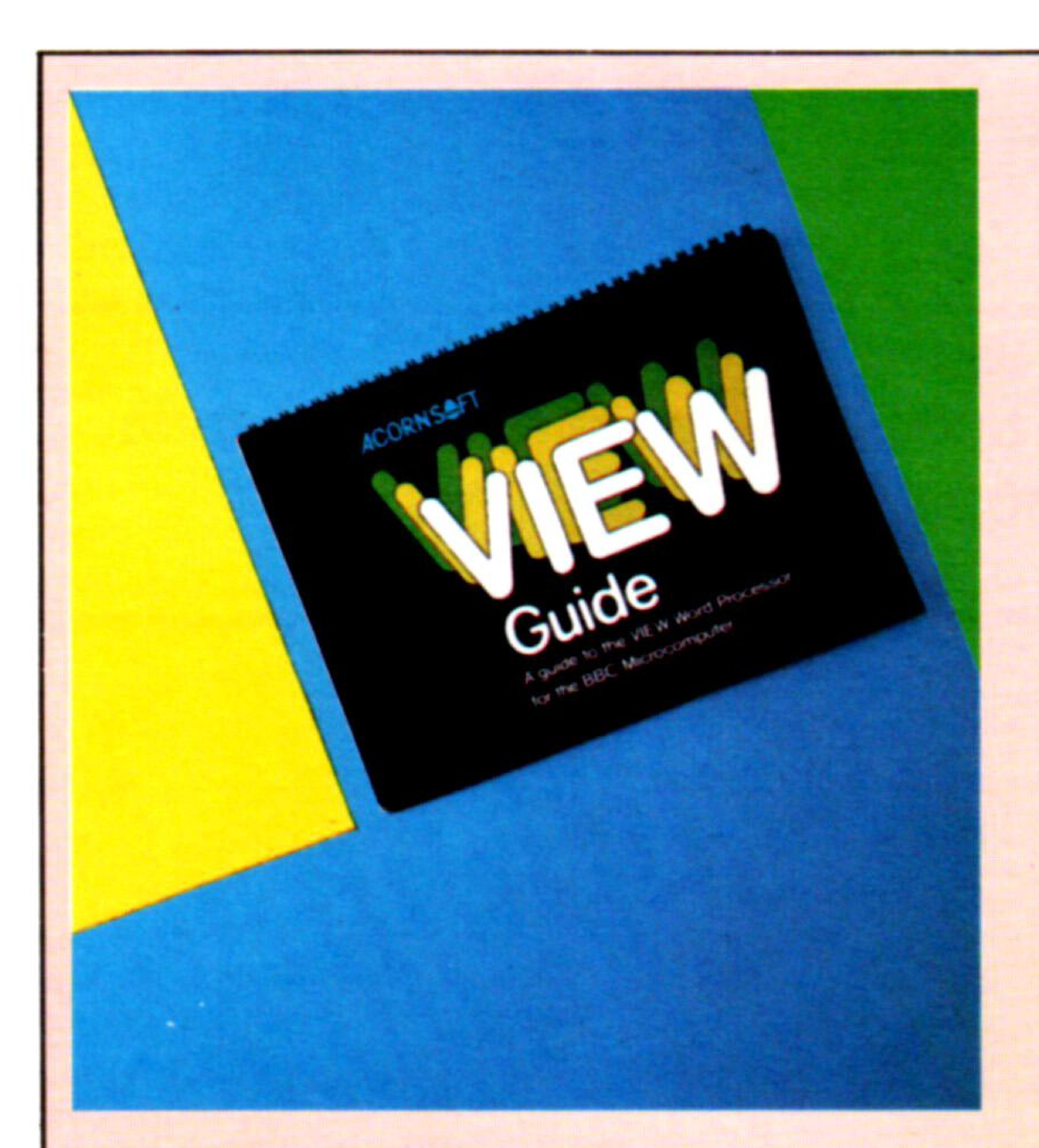
UNIDAD DE DISCO

DISCO

ÚTILES DE ESCRITORIO

MICROORDENADOR

Kevin Jones



Una “visión” alternativa

El View (“visión”), paquete de procesamiento de Acorn para el BBC Micro, también se entrega como chip de ROM. Al igual que el Wordwise, se puede instalar de manera que se cargue a sí mismo cada vez que se encienda la máquina, o se le puede dar entrada desde el BASIC.

El View utiliza la potencia del generador de video del BBC Micro para ofrecer visualizaciones de 76, 74, 34 o 16 columnas, con la consiguiente modificación de las dimensiones de los caracteres. Este efecto se produce “aumentando” la imagen completa en pantalla; la pantalla se convierte entonces en una ventana que viaja a través de la pantalla virtual, mucho más grande. La orden Mode, que efectúa estas modificaciones, opera en la modalidad Command (orden) del View, igual que las órdenes de acceso al disco y de “hallar y sustituir” e imprimir.

El recurso de “hallar y sustituir” del View, además de la función normal de buscar a través del documento cada una de las instancias en que se produce una palabra determinada, permite lo que el manual designa como *búsqueda salvaje*. La palabra se especifica de la forma normal, pero cualquier carácter dudoso se sustituye por “?” Por consiguiente, si se da entrada a “tr??” como la palabra clave, la búsqueda hallará “tras”, “tren”, “tres”, “trío”, etc., o sea, toda palabra de cuatro letras que empiece con “tr”. Hay que ser cuidadoso al definir la clave de búsqueda salvaje, pues se corre el riesgo de que el ejercicio sea contraproducente. Otra configuración del juego de órdenes permite contar las palabras de un texto.

El otro protocolo operativo del View, llamado *modalidad de pantalla (screen mode)*, se divide en dos grupos de órdenes: *immediate* (inmediatas), que controlan la inserción-supresión de caracteres, el movimiento del cursor, los movimientos de bloques y otras necesidades; y *stored* (almacenadas), que se cuidan de los parámetros para la creación o edición de un documento, dar formato a las páginas, encabezamientos y finales de páginas y otros requerimientos continuos. En la modalidad de pantalla, el View hace una utilización excelente de las teclas de función del BBC Micro, cada una de las cuales (10) posee tres usos diferentes según se empleen solas o conjuntamente con las teclas Shift o Control

Liz Heaney

Margin). Del mismo modo, LL para longitud de línea (*Line Lenght*), IN para sangrado (*INdent*), PL para longitud de página (*Page Lenght*) y así sucesivamente. Al ser mnemónicas, es fácil recordar estas órdenes y darles entrada en el texto después de pulsar una tecla de función. No aparecen en el texto visto previamente o impreso.

Si no se utilizan las órdenes definibles por el usuario, el Wordwise pasa por omisión a los valores incorporados: 70 caracteres de longitud de línea, 66 líneas de longitud de página y cinco espacios para el margen izquierdo. De modo, entonces, que se puede disponer de un documento bien trazado antes de dominar las órdenes para dar formato.

Los códigos de control de la impresora para determinar cursivas, destacar y duplicar el tamaño de los caracteres y otras opciones también se pueden digitar como órdenes definibles por el usuario.

Wordwise opera en una modalidad de menú para guardar, cargar e imprimir secciones completas de texto. El menú aparece en la pantalla cuando se selecciona Wordwise y visualiza ocho opciones. Las primeras cuatro afectan a la carga y guardado de texto en, o desde, disco o cassette.

La opción cinco le permite al usuario buscar y reemplazar ítems específicos de texto. Un ejemplo en un artículo como este que nos ocupa sería reemplazar la palabra “digitar” por “teclear”. En esta opción, las elecciones son *globales* o *selectivas*: la primera posibilita que se efectúe la sustitución en todas las instancias en las que aparece en el texto; mientras que la segunda permite que el usuario desplace el cursor hasta la primera instancia de aparición, luego a la siguiente y así sucesivamente, escogiendo si sustituir o no la palabra en cada caso. La opción seis del menú permite imprimir el texto y la siete permite verlo previamente. La opción ocho guarda el texto ya con formato sin órdenes definibles por el usuario.

Para pasar del menú a la modalidad de edición, se pulsa la tecla Escape y aparece el texto con el cursor posicionado donde se dejó. Debido a que la pantalla visualiza el texto en la modalidad de 40 columnas, con frecuencia es necesario saltar de la modalidad de edición a la opción siete (visión previa) del menú cuando se da formato al texto. Un ligero problema es que los efectos de los tabuladores no aparecen en la pantalla; por este motivo se suele requerir con frecuencia la visión previa del texto para verificar el trazado final del documento cuando se imprima.

Un aspecto interesante del BBC es que las teclas de función definidas por el usuario que utiliza el Wordwise para desplazar, copiar o eliminar texto y llevar a cabo otras tareas se pueden volver a definir para su empleo con las teclas CTRL o SHIFT con el fin, por ejemplo, de producir un nuevo párrafo, suprimir una línea completa o incluir un código para la impresora como una orden definible por el usuario. Utilizadas aisladamente, las teclas de función retienen las funciones definidas del Wordwise. Las órdenes estrella (*) del BBC se pueden utilizar en la modalidad de menú para seleccionar un tipo de impresora, seleccionar cinta o disco o retornar la máquina al BASIC digitando *B.

El Wordwise es, indudablemente, una útil adición para cualquier BBC Micro y es fácil de instalar y de utilizar. Para el BBC existen en el mercado procesadores de texto más caros y más sofisticados, como el View, pero se puede afirmar que el Wordwise satisface la mayoría de las necesidades de los usuarios.

Análisis de texto

Antes de abordar los programas de lenguaje máquina, analizaremos cómo es el área de textos en una memoria tratada con intérprete de BASIC. Saberlo nos será de gran ayuda más adelante

Cuando usted digita o carga (LOAD) un programa en BASIC en su ordenador, quizá se imagina que el ordenador es un recipiente vacío que no hace nada hasta que llegan sus instrucciones. En realidad, desde el instante mismo en que enchufa su ordenador, éste se pone a ejecutar por su cuenta un complejo programa: el sistema operativo (*Operating System*: OS). Más que un programa es un juego de programas, que está de forma permanente en algunos de los chips de ROM del interior de la máquina. Tiene como objetivo hacer que la máquina funcione: controla la pantalla, comunica con la impresora y las unidades de disco, explora el teclado en busca de pulsaciones de teclas, etc. Para el OS, cuanto entra en la máquina no es sino datos a procesar mediante sus propios programas.

Uno de estos programas se denomina *intérprete de* BASIC y tiene como objetivo inspeccionar el texto de los programas en BASIC y ejecutar las instrucciones que contienen. Por consiguiente, todo cuanto contienen nuestros programas no son más que datos a procesar por el programa intérprete. Cuando se digita un programa, el sistema operativo lo reconoce como tal porque cada nueva línea empieza con un número de línea válido. Con algunas excepciones, cada uno de los caracteres de esa línea del programa se almacena en su propio byte del área de textos de programas en BASIC de la memoria. Cuando usted digita RUN, el sistema operativo le entrega el control al intérprete de BASIC que se pone a trabajar en el procesamiento de sus datos (el contenido del área de textos en BASIC).

El intérprete no modifica su programa en ningún sentido, sino que simplemente lo interpreta y ejecuta. Y como el intérprete obedece órdenes sin más, es bastante factible darle instrucciones para que mire el contenido de cualquier área de la memoria. Si su programa le permite inspeccionar la memoria y usted lo usa para inspeccionar el área de textos, eso para el intérprete no es ninguna paradoja. Él sencillamente sigue instrucciones, si es que puede, y si no puede informa ERROR DE SINTAXIS o ERROR DE DESBORDAMIENTO DE CAPACIDAD o algo similar: no tiene ni el razonamiento ni el vocabulario para producir mensajes de error tales como: PARADOJA TEMPORAL o DISCONTINUIDAD FILOSOFICA.

El sistema operativo almacena el programa en BASIC carácter por carácter, a excepción de las instrucciones. Cada vez que reconoce las letras (o caracteres, o números, o patrones de voltaje) que componen una instrucción en BASIC, el sistema operativo reemplaza esa palabra por un número de código de un único byte, denominado *distintivo* (*token*). Así se ahorra espacio de memoria (RESTORE, p. ej., con sus 7 letras debería ocupar 7 bytes) y hace que la tarea del intérprete de traducir el programa sea mucho más fácil de llevar a cabo.

Las máquinas utilizan distintos tipos convencionales de distintivos, pero, en general, estos códigos son números mayores de 127. Los códigos ASCII para los caracteres imprimibles (reflejados en la tabla de la página siguiente) pertenecen todos a la escala entre 32 y 127. Por consiguiente, cualquier byte del área de textos en BASIC que contenga un número mayor que 127 ha de ser un byte distintivo colocado allí por el sistema operativo. Cuando el intérprete se encuentra con uno de estos bytes, ejecuta la pertinente subrutina incorporada.

Surge el interrogante, sin embargo, de por qué cuando uno LISTa un programa no ve caracteres no imprimibles sino más bien las instrucciones en BASIC. La respuesta es que durante un LISTado el sistema operativo inspecciona cada byte del área de textos y cada vez que encuentra un byte cuyo valor sea mayor que 127 lo trata como si fuera un distintivo. En algún lugar de la memoria hay almacenada una lista completa de las representaciones ASCII de las instrucciones en BASIC y el valor de un distintivo señalará dicha posición. Es lo mismo que si el intérprete estuviera utilizando el valor del distintivo para localizar la subrutina de su ejecución. Y, por consiguiente, el sistema operativo coloca en la pantalla, durante un LISTado, la palabra clave en vez del distintivo. Se lo demostrará un Commodore 64. (En el BBC y el Spectrum es menos directo.) Digite en minúsculas:

```
100 rem*******h**********
```

Ahora LISTe 100 y verá:

```
100 rem*******left$**********
```

En las máquinas Commodore el valor ASCII de la "h" en modalidad minúsculas es 200, de modo que cuando el sistema operativo encontró un valor de 200 en ese byte determinado durante el LISTado, lo interpretó como el distintivo para la palabra clave LEFT$. Si ahora digita:

```
100 rem"*******H************"
```

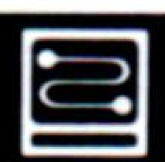
y LISTa 100, verá:

```
100 rem"*******H*************"
```

Esto demuestra que es importante recordar que algunos caracteres imprimibles, por lo general caracteres para gráficos, poseen códigos ASCII mayores que 127 y que se los reconocerá como tales siempre y cuando estén entre comillas. De no ser así, se tratarán como distintivos.

Siglas inglesas de los caracteres de control del código ASCII
Aunque los textos en castellano suelen ofrecer al lector la puntual significación de estas siglas, a cuyos caracteres el ASCII otorga valores del 0 al 31, puede que el lector desee saber de dónde derivan. He aquí su procedencia (la traducción se encuentra en el cuadro de la página siguiente)

NUL — Does nothing
SOH — Start heading
STX — Start of text
ETX — End of text
EOT — End of transmission
ENQ — Enquire
ACK — Acknowledge
BEL — Ring bell
BS — Backspace
HT — Horizontal tab
LF — Line feed
VT — Vertical tab
FF — Form feed
CR — Carriage return
SO — Shift out
SI — Shift in
DLE — Data link escape
DC1 — Device control 1
DC2 — Device control 2
DC3 — Device control 3
DC4 — Device control 4
NAK — Negative acknowledge
SYN — Synchronous idle
ETB — End of transmission block
CAN — Cancel
EM — End of medium
SUB — Substitute
ESC — Escape
FS — File separator
GS — Group separator
RS — Record separator
US — Unit separator



Llegados a este punto, podemos comenzar a investigar cómo se almacena en la memoria una línea de programa en BASIC. Los ordenadores se diferencian entre sí sólo en detalles, pero en general los primeros tres o cuatro bytes de una línea de programa en BASIC del área de textos contendrán el número de la línea del programa y alguna información relativa a la longitud de la línea (véase cuadro). El número que usted le otorga a la línea cuando la digita también se almacena (si bien no por su equivalente ASCII, pues, por ejemplo, la línea 61030 necesitaría nada menos que cinco bytes sólo para almacenar su número). Se almacena como un entero de dos bytes. De esta manera, los números entre 0 y 255 (que, recuerde, caben en un byte de ocho bits) se almacenan en un byte de ceros seguido del byte que contiene al número. Los números mayores de 255 se almacenan exactamente como las direcciones paginadas (véase p. 516): el valor del primer byte se multiplica por 256 y se le suma al valor del segundo byte. Por ejemplo, 1000 se almacenaría como 3 232 (3*256 + 232 = 1000). Estos dos bytes siempre están en la misma posición en cualquier línea almacenada en el área de textos (si bien el que sean siempre los dos primeros bytes o no depende de cada máquina).

La información relativa a la longitud de la línea se coloca en un solo byte en el BBC y en dos bytes en el Spectrum. Representa simplemente el número de bytes de la línea (incluyendo los dos bytes para el número de línea y el propio byte que contiene esta longitud). Si conoce la dirección del primer byte de una línea de programa en BASIC en la memoria, y si le suma a él el contenido del byte de longitud de línea, entonces obtendrá la dirección del primer byte de la próxima línea del programa. Dado que el mayor número que se puede expresar con un solo byte es 255, la longitud máxima de una línea de programa en BASIC en el BBC es, por consiguiente, de 255 caracteres. Usted podría utilizar el programa Mempeek de la página 539 para establecer si es ése el límite del número de *caracteres* que puede digitar en una línea de programa, o si se trata del límite de la longitud de la línea tal como ésta se almacena en el área de textos.

En el Commodore, el byte de longitud de línea se sustituye por dos bytes que se denominan *dirección de enlace*. Ésta es, simplemente, la dirección real en forma de dos bytes del primer byte de la siguiente línea del programa.

Es interesante destacar que en el BBC y en el Spectrum la dirección de comienzo de la línea siguiente se calcula a partir de la dirección presente más la longitud de línea (lo que es lento pero ahorra un byte); mientras que en el Commodore, la dirección siguiente se almacena como tal (lo cual

Diagrama de conversión
El código norteamericano estándar para intercambio de información (*American Standard Code for Information Interchange*: ASCII) proporciona a los caracteres un valor de código estándar por medio de los números entre 0 y 127. Los códigos del 0 al 31 no asumen caracteres imprimibles, sino que se utilizan para enviarles señales de control a los periféricos como la pantalla y la impresora. Por tanto, el significado de estos códigos varía considerablemente de una máquina a otra, tal como refleja el diagrama. Algunas máquinas, en especial el Commodore y el Spectrum en nuestro diagrama, dejan muchos códigos sin utilizar (señalados aquí mediante un ●). Los códigos del 32 al 127 son para los caracteres imprimibles; los códigos ASCII estándar dentro de esta escala son los más generalizados (con variaciones menores) en la mayoría de los ordenadores. Su manual para el usuario le proporcionará los códigos ASCII para su máquina

CODIGO ASCII	ASCII	COMMODORE	SPECTRUM	BBC MICRO
0	NUL — No hace nada	●	●	Nulo
1	SOH — Inicio de encabezamiento	●	●	Siguiente carácter a impresora
2	STX — Inicio de texto	●	●	Activar impresora
3	ETX — Fin de texto	●	●	Desactivar impresora
4	EOT — Fin de transmisión	●	●	Separar cursores texto-gráficos
5	ENQ — Consulta	Tecla blanco	●	Juntar cursores texto-gráficos
6	ACK — Acuse de recibo	●	PRINT	Activar unidades VDU
7	BEL — Sonar timbre	●	EDIT	Emitir pitido breve
8	BS — Retroceder un espacio	Desactiva tecla CBM	Cursor izq.	Cursor atrás un espacio
9	HT — Tabulación horizontal	Activa tecla CBM	Cursor der.	Cursor adelante un espacio
10	LF — Avance de línea	●	Cursor abajo	Cursor abajo
11	VT — Tabulación vertical	●	Cursor arriba	Cursor arriba
12	FF — Salto vertical	●	Delete	Limpiar área de texto
13	CR — Retorno del carro	RETURN	ENTER	Return
14	SO — Desplazar hacia afuera	Encender minúsculas	Número	Encender modalidad página
15	SI — Desplazar hacia adentro	●	●	Apagar modalidad página
16	DLE — Escape enlace de datos	●	INK	Limpiar área gráficos
17	DC1 — Control del dispositivo 1	Cursor abajo	PAPER	Def. color texto
18	DC2 — Control del dispositivo 2	Reverse on	FLASH	Def. color gráficos
19	DC3 — Control del dispositivo 3	Cursor a base	BRIGHT	Def. color lógico
20	DC4 — Control del dispositivo 4	Delete	INVERSE	Restaurar ausencia color lógico
21	NAK — Recibido negativo	●	OVER	Desactivar unidades VDU
22	SYN — DLE sincrónico	●	AT	Seleccionar modalidad pantalla
23	ETB — Fin bloque transmisión	●	TAB	Reprogramar carácter visualiz.
24	CAN — Cancelar	●	●	Def. ventana gráficos
25	EM — Fin del medio	●	●	Plot m,x,y
26	SUB — Sustituir	●	●	Restaurar ausencia ventanas
27	ESC — Carácter de escape	●	●	Nulo
28	FS — Separador de archivos	Tecla rojo	●	Def. ventana textos
29	GS — Separador de grupo	Cursor der.	●	Def. origen gráficos
30	RS — Separador de registros	Tecla verde	●	Desplazar cursor texto
31	US — Separador de unidades	Tecla azul	●	Desplazar cursor texto a x,y

CODIGO ASCII	ASCII	CODIGO ASCII	ASCII
32	Espacio	80	P
33	!	81	Q
34	"	82	R
35	#	83	S
36	$	84	T
37	%	85	U
38	&	86	V
39	'	87	W
40	(	88	X
41	)	89	Y
42	*	90	Z
43	+	91	[
44	,	92	\
45	–	93	]
46	.	94	^
47	/	95	_
48	0	96	|
49	1	97	a
50	2	98	b
51	3	99	c
52	4	100	d
53	5	101	e
54	6	102	f
55	7	103	g
56	8	104	h
57	9	105	i
58	:	106	j
59	;	107	k
60	<	108	l
61	=	109	m
62	>	110	n
63	?	111	o
64	@	112	p
65	A	113	q
66	B	114	r
67	C	115	s
68	D	116	t
69	E	117	u
70	F	118	v
71	G	119	w
72	H	120	x
73	I	121	y
74	J	122	z
75	K	123	{
76	L	124	!
77	M	125	}
78	N	126	"
79	O	127	Delete

entraña la utilización de un byte extra pero es más rápido). Esto demuestra que no hay ninguna forma *correcta* de construir un ordenador, sólo existe la manera en que lo hace el diseñador individual. También constituye un buen ejemplo de las cosas que los diseñadores de ordenadores deben tener en cuenta. Ellos saben que tienen que hacer una elección fundamental entre diseñar una máquina que sea lenta pero barata, o una que sea rápida pero cara. Del mismo modo, al escribir un programa en BASIC en máquinas con memoria limitada (el Vic-20 y ZX81 sin ampliar constituyen buenos ejemplos de ello), se debe encontrar un equilibrio entre velocidad de ejecución y eficacia en el uso de la memoria.

Por último, observe que en el área de textos habrá un indicador de comienzo de línea o final de línea para cada línea del programa en BASIC. En el BBC Micro cada línea empieza con un byte que contiene 13 (ASCII para Retorno de carro), mientras que en una línea del Spectrum éste se halla al final. En el Commodore, la línea de BASIC termina con un byte-cero (el ASCII de "").

Cómo se almacenan los programas en BASIC

Cada máquina posee sus variantes propias en cuanto a la forma en que almacena una línea de texto en BASIC. Observe las técnicas individuales para las líneas de texto que proporcionamos abajo

BBC Micro

200 PRINT "CADENA ASCII"
300 A = 1963.2:B = INT(A):A$ = "C"

contenido de los bytes de memoria

13	0	200	20	32	241	34	67	65	68	69	78	65	32	65	83	67	73	73	34
CR	número de línea		número total de bytes de la línea	esp	distintivo PRINT	"	C	A	D	E	N	A	esp	A	S	C	I	I	"

indicador comienzo de línea

En este ejemplo, las palabras clave en BASIC se sustituyen por distintivos de un solo byte. Todos los otros caracteres se almacenan como códigos ASCII. Los bytes del indicador de comienzo de línea, de número de línea y de longitud de línea los pone automáticamente el sistema operativo

13	1	44	27	65	61	49	57	54	51	46	50	58	66	61	168	40	65	41	58	65	36	61	34	67	34
CR	número de línea		número total de bytes de la línea	A	=	1	9	6	3	.	2	:	B	=	distintivo INT	(	A	)	:	A	$	=	"	C	"

indicador comienzo de línea

Commodore 64

200 PRINT "CADENA ASCII"
300 A = 1963.2:B = INT(A)

240	9	200	0	153	34	67	65	68	69	78	65	32	65	83	67	73	73	34	0
dirección enlace		número de línea		distintivo PRINT	"	C	A	D	E	N	A	esp	A	S	C	I	I	"	marcador de final de línea

La dirección de enlace proporciona la dirección del primer byte de la línea siguiente. Observe también que los bytes de la dirección de enlace y del número de línea están en forma de byte de desplazamiento seguido de byte de página

4	10	44	1	65	178	49	57	54	51	46	50	58	66	178	181	40	65	41	0
dirección enlace		número de línea		A	distintivo '='	1	9	6	3	.	2	:	B	distintivo '='	distintivo INT	(	A	)	marcador de final de línea

Spectrum

200 PRINT "CADENA ASCII"
300 LET A = 1963.2:LET B = INT A

0	200	16	0	245	34	67	65	68	69	78	65	32	65		83	67	73	34	13
número de línea		longitud de la línea del programa más byte fin de línea		distintivo PRINT	"	C	A	D	E	N	A	esp	A	S	C	I	I	"	CR (indicador fin de línea)

Observe que la longitud de la línea se expresa en dos bytes en vez de en uno, de modo que las líneas del programa de longitud mayor que 255 caracteres son factibles. Además, observe que la constante numérica 1963.2 se almacena primero en códigos ASCII y luego en un formato binario especial. Ello mejora la velocidad de ejecución del programa

1	44	22	0	241	65	61	49	57	54	51	46	50	14	139	117	102	102	102	58	241	66	61	186	65	13
número de línea		longitud de la línea del programa más byte fin de línea		distintivo LET	A	=	1	9	6	3	.	2	1963.2 en formato binario						:	distintivo LET	B	=	distintivo INT	A	CR (indicador de final de línea)



La fórmula del éxito

Creada a fines de 1982, Imagine Software ha sabido aprovechar el auge de los ordenadores personales. Del centenar de empleados con que cuenta, más de cuarenta escriben código de programación

Tal como ocurre en una empresa que triunfa, Imagine debe su éxito en el terreno del software a una mezcla de talentos. Los fundadores, David Lawson y Mark Butler, ambos de Liverpool, representan los dos elementos esenciales de cualquier empresa: la experiencia técnica y la agudeza comercial.

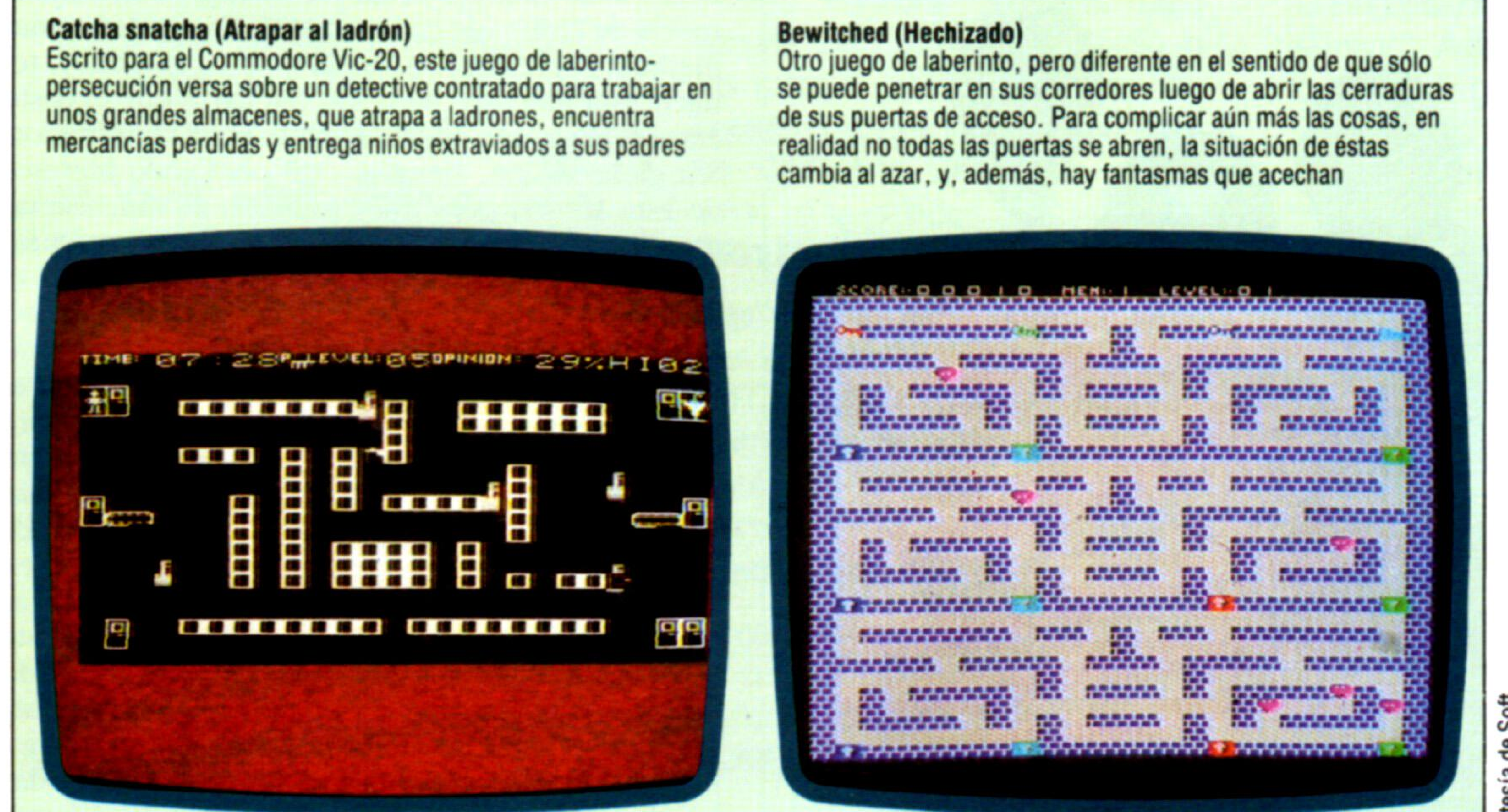
La especialidad de Imagine consiste en escribir software para las máquinas Commodore. En un momento dado, cuatro programas para juegos distintos (*Bewitched* [Hechizado], *Catcha snatcha* [Atrapar al ladrón], *Wacky waiters* [Locos camareros] y el tradicional *Arcadia*) escritos por Imagine para Commodore figuraban en la lista de los diez programas más vendidos. Los esfuerzos que Imagine dedicó a los ordenadores de la Commodore motivaron en 1983 la insólita distinción de ser invitada por ésta a que visitara la fábrica y las oficinas de Norristown (Pennsylvania) para que conociera, antes de su salida al mercado, las máquinas 264 y V364. Estas máquinas están destinadas a los amantes de los juegos y hasta ahora Commodore ha resistido a la tentación de elevar a sus productos en la escala de precios, con el fin de atraer al usuario pequeño empresario. A raíz de la gira, Commodore encargó a Imagine dos nuevos juegos, que se venderán bajo su propio cuño, lo que representa todo un triunfo para una empresa tan joven.

En Gran Bretaña, sin embargo, el que disfruta del apoyo de software más amplio es el Sinclair Spectrum. Pero las relaciones de Imagine con sir Clive Sinclair no han sido siempre inmejorables. Cuando Mark Butler y Dave Lawson se marcharon de Bug-Byte Software, donde habían pasado sus años de formación, programaron una visita a Sinclair, pero ésta acabó sin llegar a acuerdo alguno de trabajo. La política de Imagine es que sus juegos se puedan modificar para adaptarse a una nueva máquina sin que para ello se requiera mucho más que cambiar las posiciones de memoria de unos pocos trozos de código del programa.

El QL se basa en el procesador central Motorola 68000 y actualmente la mayoría de los programadores que poseen experiencia profesional con este chip ya están contratados por firmas profesionales de software o trabajan en los departamentos de informática de las universidades.

Con el fin de encontrar programadores para que trabajaran con el procesador 68000, Imagine puso anuncios para cubrir un buen número de puestos, pero en líneas generales los resultados fueron decepcionantes. Aunque hubo una masiva respuesta, eran muy pocas las personas cuya experiencia fuera superior a unos pocos meses. Muchas ni siquiera habían logrado completar la codificación de un solo juego. ¡No había materia prima para iniciar una nueva generación de proyectos de programación!

Parece ser que Imagine está ahora planeando una nueva estrategia de cara al mercado más lucrativo (y más competitivo) del software para gestión empresarial. En realidad no se trata tanto de una



Catcha snatcha (Atrapar al ladrón)
Escrito para el Commodore Vic-20, este juego de laberinto-persecución versa sobre un detective contratado para trabajar en unos grandes almacenes, que atrapa a ladrones, encuentra mercancías perdidas y entrega niños extraviados a sus padres

Bewitched (Hechizado)
Otro juego de laberinto, pero diferente en el sentido de que sólo se puede penetrar en sus corredores luego de abrir las cerraduras de sus puertas de acceso. Para complicar aún más las cosas, en realidad no todas las puertas se abren, la situación de éstas cambia al azar, y, además, hay fantasmas que acechan

Cortesía de Soft

táctica original, porque fue Apple quien inició los primeros contactos. Aunque en un principio la iniciativa sorprendió a Imagine, ésta pronto comprendió por dónde Apple estaba dirigiendo su software, con una representación en pantalla fácil de utilizar e imágenes pictóricas como "ventanas" e "iconos" enlazadas mediante el dispositivo de entrada del ratón, en donde la acción gráfica y la entrada no efectuada desde el teclado constituyen factores clave.

Imagine tenía un sólido conocimiento de la tecnología en ocho bits de Apple, porque había utilizado ordenadores Apple IIe para escribir programas desde el principio. En consecuencia, estaba en condiciones ideales de sacar partido de este contacto. Asimismo, como parte de un ejercicio de renovación instrumental, Imagine invirtió en varias máquinas Sage IV. El objeto de ello fue el de aumentar la velocidad y el potencial informático. Dave Lawson ha escrito bastante software original utilizado, a su vez, para escribir programas de juegos, que luego se compilan de forma cruzada (se compilan en una máquina y se emplean en otra) para el ordenador personal en cuestión. El Sage IV posee un disco de RAM de gran capacidad, cuyo valor no tiene precio para escribir en lenguaje ensamblador. De cara al tiempo de los programadores, es más económico gastar dinero en una máquina como el Sage, que tener a los programadores sentados mientras esperan programas compilados. Junto con el Sage se proporciona un p-System UCSD, **U**niversidad de **C**alifornia en **S**an **D**iego, una implementación del PASCAL para imitar al Lisa y al Mackintosh de Apple, que utilizan ambos PASCAL en el *back ground* (zona de baja prioridad en la memoria). Los términos *background* y *foreground* se aplican a máquinas que pueden ejecutar más de un programa a la vez. El programa "foreground" siempre tiene prioridad, pero cada vez que se presente una oportunidad se ejecutará el programa "background".

Como muchas otras casas de software, Imagine ha estado investigando el lenguaje para ordenadores que lleva el nombre más conciso: el C. Éste es uno de los lenguajes más versátiles y, aun más importante, más portátiles de todos los que existen para microordenadores. Su estructura modular lo hace ideal para desarrollar software de sistemas.

En la actualidad no existen muchas personas que posean experiencia tanto en C como en el microordenador Sage. Por eso Imagine está rastreando en las facultades universitarias y en otras casas de software, con el deseo de hallar más personas con experiencia en este campo.

Cuarteto "imaginativo"
La pujanza de cualquier firma de software radica en la composición adecuada del equipo de diseño de programas. Unidos específicamente para crear los juegos Psyclapse y Bandersnatch, en la foto aparecen, de izquierda a derecha: Ian Weatherburn, Mike Glover, John Gibson y Eugene Evans

Cortesía de Imagine

El secreto del triunfo

¿Qué es lo que hace que un programa se convierta en un éxito? Mucho tienen que ver en ello las "corazonadas". Al principio, cuando Mark Butler y Dave Lawson juntaron sus fuerzas para lanzar Imagine, ya tenían una clara idea de lo que la firma donde habían trabajado, Bug-Byte Software, estaba comercializando con todo éxito. Se sentaban y se sometían a sesiones para inspirarse y crear ideas: intentaban ponerse mentalmente en el lugar del entusiasta de los juegos. Lo que es primordial. "Si éste fuera mi primer ordenador, que tendré por un mes, o por seis meses, ¿qué esperaría de este juego? ¿Por qué iba yo a jugar con él? ¿Durante cuánto tiempo jugaré? En un punto del juego, ¿me gustarán los diversos efectos sonoros o un flash de los gráficos?" El diseñador de juegos tiene que pensar como si él mismo tuviera la edad del cliente al que se destina el juego.

Ahora Imagine es demasiado grande como para que dos personas solas generen todas las ideas para los juegos, y actualmente cuenta con un equipo de ocho diseñadores de gráficos que trabajan en la animación y secuencia gráfica de las nuevas obras. Éstas luego son probadas por el personal de la casa, no por los programadores.

Por desgracia para las casas de software, una técnica que han aprendido demasiadas personas sin conocimientos específicos es cómo ahorrarse el desembolso por el software. La reproducción ilegal es un problema acuciante. Se ha calculado que por cada juego en cassette que se vende desde el mostrador se hacen siete copias ilegales, la inmensa mayoría de ellas por el simple trámite de la transferencia de audio (de cinta a cinta). Se pueden tomar medidas preventivas para eliminar o, al menos, reducir esta práctica, pero son tan caras que el costo extra repercutiría negativamente en el consumidor. Sin duda alguna, Imagine está perdiendo ingresos de esta forma, pero así y todo vende muchísimas cassettes y por el momento mantiene sus opciones.

Puede que no pase mucho tiempo antes de que el cartucho y el disco sustituyan por completo a la cassette. Incluso podrían surgir procedimientos completamente nuevos para la distribución del software, en los cuales el código del programa se enviara desde un ordenador central, o anfitrión, y se cargara en el ordenador personal a través de las líneas telefónicas que se utilizan para la televisión por cable. Una firma de software que aspire a crecer en la década actual tiene necesariamente que estar atenta a las modificaciones de este tipo y, en este sentido, Mark Butler, de Imagine, está poniendo todo su talento al servicio de ello, sugiriendo nuevas posibilidades para los juegos empleando la síntesis de voz y manteniéndose al corriente de todas las innovaciones tecnológicas, como el disco láser.

PARA JUGAR A LO GRANDE (INSTANTANEAMENTE)

Presentamos **el Interface 2 ZX** Pensado y diseñado por SINCLAIR para unirse a la perfección con tu microordenador Spectrum.

Si a la hora de elegir tu microordenador optaste por el mejor, es lógico que elijas ahora el Interface 2 ZX

Ya habrás podido deleitarte con la más amplia variedad de juegos existentes para tu Spectrum (la más extensa del mercado). Ahora con el Interface 2 ZX vas a tener más ventajas para tu Spectrum:



- Podrás conectar Joysticks para sacarle, aún, mayor rendimiento a tus mejores juegos y divertirte con aquellos exclusivamente disponibles en **Cartuchos ZX:** correr, saltar, volar... a lo grande.
  ¡Menuda diferencia!
- Además, al ser cartuchos con memoria ROM, podrás, con tu SPECTRUM de 16 K, jugar con programas hasta ahora reservados para 48 K, sin ampliar la memoria.
  ¡Vaya ahorro!
- Al conectar el Interface 2 ZX tienes la certeza de poseer un periférico pensado por SINCLAIR para SINCLAIR. Tu microordenador queda a salvo de circuitos poco fiables.
  ¡Un alivio!
- Al adquirir el Interface 2 ZX y los Cartuchos ZX en la red de Concesionarios Autorizados, podrás exigir la tarjeta de garantía INVESTRONICA, única válida en territorio nacional.
  ¡Una tranquilidad!

Interface 2 ZX y Cartuchos ZX

Si aún no los tienes
no sabes lo que te pierdes

Solicita una demostración en cualquier Concesionario Autorizado INVESTRONICA.



9 788485 822836 10028

